

Final Project



Reykjavik University
Computer science

Tengir - Service Portal

Arnar Björn Pálsson

Atli Egilsson

Viðar Einarsson

Instructor: Hildur Andriksdóttir

Examiner: Stefán Freyr Stefánsson

1 Introduction	4
2 Organization	5
2.1 Roles	5
2.2 Communication	5
2.2.1 Meetings with instructor	6
2.2.2 Meetings with client	6
2.2.3 Team meetings	6
2.2.3.1 Sprint plan tracking	6
2.2.3.2 Sprint retrospective	7
2.3 Technical Environment	7
2.4 Physical environment	8
3 Work Schedule	9
3.1 Sprints	9
3.2 Capacity plan	10
3.3 Release dates	11
4 Risk Analysis	12
4.1 Projects top risks	12
5. Requirement Analysis	16
5.1 User groups	16
5.2 Requirement list	17
5.2.1 Functional requirements	17
5.2.2 Non functional requirements	19
5.3 Evaluation of websites	19
5.4 Interviews	20
5.4.1 About the interviewees	20
5.4.2 Interpretation of interviews	21
5.4.2.1 Employees	21
5.4.2.2 Customers	21
5.5 Use cases	22
5.6 Results	29
5.6.1 Requirements added	29
5.6.2 Requirements changed/removed	30
6 System Design	31
6.1 Page prototypes	31
6.2 Connection to other systems	36
6.3 Workflow	37
6.3.1 Navigation diagrams	37
6.3.2 Website flow	38

6.3.2.1 User settings	38
6.3.2.2 Printing / Emailing	39
6.3.2.3 TV connection	40
6.4 Coding rules	41
6.4.1 General programming rules	41
6.4.2 C# / JavaScript code rules	41
6.4.3 HTML	43
6.4.4 CSS	43
6.5 Database	44
6.5.1 Rules	44
6.6 Testing	44
6.7 Websites style	45
6.8 Solution	46
7 Progress overview	47
7.1 Organization Plan [24. Jan]	47
7.2 Design and Analysis [7. Feb]	48
7.3 Prototype [11. Mar]	49
7.4 Final product [11. May]	50
8 Project retrospective	51
8.1 What went well	51
8.2 What went badly	51
8.3 What the team would do differently	52
8.4 Future work	52
8.4.1 SNMP connection	52
8.4.2 Area addon	52
8.4.3 Electronic ID	53
8.4.4 Integrating other systems	53
8.4.5 Request forms	53
8.5 Special thanks	54
Appendix A	55
A.1 Sprint Zero [18.jan - 24.jan]	55
A.1.1 Retrospective	55
A.1.2 Sprint backlog & burndown chart	56
A.2 Sprint 1 [25.jan - 7.feb]	57
A.2.1 Retrospective	57
A.2.2 Sprint backlog & burndown chart	57
A.3 Sprint 2 [8. feb - 25. feb]	59
A.3.1 Retrospective	59
A.3.2 Sprint backlog & burndown chart	60
A.4 Sprint 3 [26. feb - 11. mar]	61

A.4.1 Retrospective	61
A.4.2 Sprint backlog & burndown chart	62
A.5 Sprint 4 [12. mar - 13. apr]	63
A.5.1 Retrospective	63
A.5.2 Sprint backlog & burndown chart	64
A.6 Sprint 5 [16. apr - 22. apr]	65
A.6.1 Retrospective	65
A.6.2 Sprint backlog & burndown chart	66
A.7 Sprint 6 [23. apr - 29. apr]	67
A.7.1 Retrospective	67
A.7.2 Sprint backlog & burndown chart	68
A.8 Sprint 7 [30. apr - 6. May]	69
A.8.1 Retrospective	69
A.8.2 Sprint backlog & burndown chart	70
A.9 Sprint 8 [7. may - 14. May]	71
A.9.1 Retrospective	71
A.9.2 Sprint backlog & burndown chart	72
Appendix B	73
B.1 Interview questions	73
Appendix C	74
C.1 Glossary	74
Appendix D	75
D.1 Manuals	75

1 Introduction

Tengir is a local company that specializes in telecom solutions for companies and individuals in the northeastern part of Iceland. They have been growing at a high rate and so has their user group which results in more traffic and service. By designing a service portal for their customers, a lot of simple tasks can be automated that are now being handled through phone calls and emails. The way it is handled now, is that customers have to call the offices of Tengir to make any kind of request, e.g. getting an overview of their invoices for the month. These invoices then have to be emailed or printed out and sent to the customer. The aim is for this system to reduce these number of calls and the paper that is being wasted by developing a website where customers of Tengir can log in and view their invoices, subscriptions and more.

Every aspect of the design and development of the web application will be thoroughly documented and used as a guideline for a well organized project and workflow. This report will help you understand the team structure and what methods the team will use for development. Most impactful risks will be laid out along with how they will be treated and the requirements for the application will be analyzed. Some prototypes of the design and structure of the code will be presented. Lastly, an overview of the teams progress throughout the project is shown with an insight on how things were handled. Please refer to the [glossary](#) provided in appendix C for information on the abbreviations used in the report.

2 Organization

While working on the project, the team will use the Scrum methodology. It was a group decision to go with Scrum due to how agile it is and easy to work with. It really helps to be able to customize the sprints according to workload at school.

Previous experience was also factored in and all members had finished the Software Engineering course taught at RU, where Scrum was used. During that course each member learned how to set up and start working with Scrum and it worked out great. This chapter will go through few things about Scrum and decisions regarding it, including sprint lengths, Scrum roles, communication and more.

2.1 Roles

The project owners are the CEO, Gunnar Björn Þórhallsson, and main developer, Ingvar Karl Þorsteinsson, at Tengir hf. During the work the team will definitely get a lot of input for the system from Gunnar, both looks and features, while also receiving insights from Ingvar. It was decided that Viðar would take the role of the Scrum master since he has the most connection within Tengir and that could help prepare for daily meetings, meetings with the Project Owner and more. Since the team is relatively small and only consists of three members, the scrum master will also be a member of the development team.

Project Owner	Tengir hf. [Gunnar Björn and Ingvar Karl]
Scrum Master	Viðar Einarsson
Development Team	Arnar Björn Pálsson, Atli Egilsson and Viðar Einarsson

Table 1: The scrum roles

2.2 Communication

It is really important, both for the development team and the client, that communication is on a regular basis, that is the reason for a guideline that the team will follow throughout the project. Meetings are arranged for the team, with the instructor and client, separately.

2.2.1 Meetings with instructor

Early in development, every monday at 13:00, a meeting is held with the projects instructor, Hildur. These sessions will be used to evaluate ideas and consult her with new ones. Earlier meetings are expected to be more useful than the later ones since the team will mostly be programming later on. The later meetings might be cancelled if the team is just programming and have no questions or concerns to consult her about. Her insight on the design and all preparation will be highly valuable, therefore each meeting will be documented for future references.

2.2.2 Meetings with client

There will be two kinds of meetings with Tengir, one for support and one for consulting. Whenever there are any issues regarding code, they will be noted down and brought up with Ingvar. These support meetings will be held on a weekly basis and will not start until the programming begins. He will be around the workplace and can show up with a short notice off-schedule.

At the end of each sprint, a retrospective meeting will be held and there will be discussions about what was accomplished, and what wasn't. Next sprint will also be presented where ideas and speculations will be brought up and discussed with Gunnar, the Product Owner. Gunnar will give his input on the progress so far and concerns, if any.

2.2.3 Team meetings

The following subchapters will cover the various team meetings which will be held throughout the project. These meetings include daily Scrum meetings held every morning and sprint retrospective meetings are held after every sprint.

2.2.3.1 Sprint plan tracking

At the beginning of each day the team will gather together at the office where a stand-up Scrum meeting will take place. These meetings will be used to talk about accomplishments the previous day, what is planned for today and concerns regarding any issues in the way. This helps everyone stay on the same page and avoid conflicts in the group. Ideas can be shared regarding any problems, and thus, lifting the spirit for the upcoming work day.

2.2.3.2 Sprint retrospective

Before a new sprint is started, the team will reflect on the previous one and share their opinions on how it went and what could have gone better. The tasks from the sprint backlog will be viewed individually and the team decides if they will be cancelled or continued. When those matters have been discussed, new tasks and tasks from the backlog will be put in the next sprint backlog.

2.3 Technical Environment

Integrated development environment

The IDE used for this project is Visual Studio 2017, and in such a big project there are a lot of useful programs to help with development other than the IDE itself.

Project management

An online agile project management software called Taiga.io will be great for keeping track of sprints. It allows you to create new sprints, assign members to product backlog tasks and many more useful features.

Version control system

The code will be hosted on Github in a private repository and shared between the group. The main reason that the team went with Github instead of other version control systems was due to the fact that every single team member had the most experience with Github and had access to private repositories.

Documentation

To keep track of the work effort by each team member on the project, a spreadsheet will be hosted on Google Drive where every hour is logged with details on what the time is spent on.

Diagrams & prototypes

For diagrams, an online tool called LucidChart will be used. The web-application offers a modern interface that can be used to create most types of diagrams and team members can collaborate on each and every one. Balsamiq was also used to create the system prototypes that can be viewed in the [Design chapter](#).

Slack

The team will take advantage of the services Slack has to offer while in development. It has been integrated with Taiga so whenever a new task is added or moved from e.g. In Progress to Done, it will be automatically posted on the corresponding Slack channel. A general channel will also be used for sharing ideas and discussions.

2.4 Physical environment

The offices of Tengir will be provided for the project where the team will have great working conditions. There will be access to a meeting room, where the daily scrum meetings and general work on the project will be handled. The room has a big whiteboard which will be useful for scrum meetings.

Access to the offices is unrestricted since Viðar will have a keycard which gives the team access whenever needed. There will also be access to the entire staff so getting feedback or help from them is easily sought.

3 Work Schedule

Following the Scrum protocol, the project period is divided into sprints and their deadlines are aligned with release dates of products and reports. Working conditions and capacity is useful for planning ahead and making sure time is not wasted. To have a feeling for the work effort possible, the team has an estimate as to how much work each member can provide.

3.1 Sprints

The following table [Table 2] shows the sprints for the project. The table has been updated regularly throughout the project to match their more detailed chapter in the [appendix](#).

Sprint number	Sprint length	Sprint goal
Sprint 0	18.Jan - 24.Jan	Draft of work schedule Report on organization planning General preparation
Sprint 1	25.Jan - 7.Feb	Complete work schedule Design Risk analysis Requirement analysis Progress overview
Sprint 2	8.Feb - 25.Feb	Requirement analysis Design Set up technical environment Prepare for programming work
Sprint 3	26.Feb - 11.March	Connect to the inner system Register as a new user Setup system database Update final report
Sprint 4	12.March - 13.April	Update final report Planning & Meetings with instructor and developer

Sprint 5	16.April - 22.April	Connect to accounting system Users current invoice User changes account information
Sprint 6	23.April - 29.April	User connection User requests / cancels service User invoice search
Sprint 7	30.April - 6.May	Finish programming main features Operation manual User manual
Sprint 8	7.May - 14.May	Final product Finish documentation and final report Presentation preparation

Table 2: Sprints and their length

3.2 Capacity plan

The following table [Table 3] shows the team's work capacity. There are different work schedules for each type of period. Reason being that the work load is heavier at school before exams and the project will be treated as a full-time job after. As seen in the following table [Table 3], there are three different periods while working on this project. The normal period shows how much time will be spent before the exams where the project will be worked on alongside three other courses. The exam period shows how much time will be spent on it while the team prepares for the exams and final projects in other courses. Finally, the full-time period represents the workflow after the exams where it will be treated as a full time job.

The capacity plan assumes that weekends are days off but it's safe to assume that some time will be spent there, especially near deadlines.

	Time period	Team capacity	Team member capacity
Normal period	10. Jan - 11. Mars	60 hours per week	20 hours per week
Exam period	12. Mars - 15. April	30 hours per week	10 hours per week
Full time period	16. April - 11. May	120 hours per week	40 hours per week

Table 3: Project periods

3.3 Release dates

There are numerous reports which are going to be written along the way which will help analyse and prepare for programming work. The following table [Table 4] shows which reports will be made and their due date.

Product	Release date
Organization planning Work schedule draft	January 24th
Work schedule, Risk analysis, Design draft, Progress overview, Requirement analysis	February 7th
Progress overview	March 11th
Operation manual	May 7th
Final product	May 11th

Table 4: Release dates of products

4 Risk Analysis

A risk analysis was performed to get a better overview of what could go wrong in the implementation of the project, and how much exposure it would leave if it did. By doing this it is possible to lower the chance of anything going wrong since discussing how to avoid these risks can help tremendously, how to deal with failures and manage time more efficiently resulting in a better outcome overall.

4.1 Projects top risks

The projects top risks and more common risks were identified and put in the following table [Table 5]. The probability that the risk will happen and the loss of time from it are taken into consideration when calculating the exposure of the risk.

Each risk has a number depending on the probability of it happening and then again depending on the impact it will have on the project so it's possible to prioritize the risks based on those factors. The exposure is then calculated as a product of the probability and size of loss. Each member of the team gets assigned to deal with a risk depending on his experience and knowledge or his availability. The accounting and inner systems are not under the team's control, and if anything happens to them, Ingvar the main programmer at Tengir will be in charge of dealing with them. It was decided that Viðar would be his main contact in case that happens. Due to the size of the table it was decided to split it up into three parts but should be viewed as a single table.

#	Risk	Probability (1-5)	Impact (1-5)	Exposure	Charge	Prevent from happening	Minimize loss
1	Workload in other courses gets too heavy	5	3	15	Arnar	Organize the semester and plan ahead.	Team up in other courses to work faster.
Event log for #1		<p>15/02/18: Big assignment in a course takes a lot of time. The team teams up for that project to finish it faster.</p> <p>20/03/18: Three other courses have final projects. The team teams up in these projects to finish them faster, meanwhile everything is set on hold in this project.</p> <p>10/04/19: The team is in final exams, meanwhile everything is set on hold in this project.</p>					
2	Team member takes a vacation.	3	2	6	Viðar	Try not to plan vacation at busiest time in the semester.	Work extra hours before vacation to meet deadlines.
Event log for #2		<p>25/02/18: Viðar goes to the UK. He works extra hours before leaving to finish his deadlines.</p> <p>03/02/18: Atli goes to Reykjavík. He works extra hours before leaving to finish his deadlines.</p>					
3	The framework doesn't directly support systems DBs	2	5	10	Viðar	Do research before choosing framework to make sure every aspect of the project is supported	Mock data to work with while working on a solution.
Event log for #3		<p>05/03/18: Access database is not directly supported in the projects framework. While working on a solution the team exported the data to a SQL database to be able to work with it. In cooperation with Ingvar, the team found a NuGet package which supports it.</p>					

#	Risk	Probability (1-5)	Impact (1-5)	Exposure	Charge	Prevent from happening	Minimize loss
4	UI not compatible across major browsers	2	3	6	Atli	Test across all major browsers. Make sure CSS is supported by using caniuse .	Make sure that the most important part of the UI will still be visible to the user.
Event log for #4		During development, the website was tested on Google Chrome and Edge with no problems.					
5	Illness in team resulting in failed deadlines	2	2	4	Arnar	Unfortunately, there is no way to prevent this from happening.	Distribute the tasks between the other developers.
Event log for #5		10/03/18: The whole team caught a cold for a couple of days. Even though not all team member showed up they still could contribute to the project from home, although the work speed suffered a bit.					
6	Inner system crashes.	1	5	5	Viðar	Unfortunately, server crashes can happen suddenly and unannounced.	Display a mock of the data along with a message saying it might not be the newest data.
Event log for #6		Fortunately, this never happened.					

#	Risk	Probability (1-5)	Impact (1-5)	Exposure	Charge	Prevent from happening	Minimize loss
7	Accounting system crashes.	1	5	5	Viðar	Unfortunately, server crashes can happen suddenly and unannounced.	Display a mock of the data along with a message saying it might not be the newest data.
Event log for #7		Fortunately, this never happened.					
8	Team members computer crashes.	1	1	1	Atli	Unfortunately, computers are a hardware which can crash and fail without any warning.	Commit code regularly. Reports and timesheed stored on Google docs. Tengir can provide a computer temporarily if needed.
Event log for #8		Fortunately, this never happened.					

Table 5: Risks and their details

5. Requirement Analysis

Everything of high importance that is necessary for the system to work as a whole will be covered in this chapter. Users are categorized and resources sought from various places in order to improve the design of the system.

While working on the Requirement Analysis it was decided to evaluate other similar websites to get a better feeling on how it could possibly look and function. The system user groups were also analyzed and a requirement list was created to get a better overview of the system requirements and how they affect the user groups. Interviews were conducted with staff of Tengir to find out which requests customers most frequently ask and possibly get feedback on something that was missed in the making of the requirement list.

5.1 User groups

The following table [Table 6] shows which user groups belong to the system. After analyzing the system the user groups that were decided upon are the following, normal customers and businesses. These user groups will be very similar but their main goals might vary a bit from user to user.

User group	Background	System usage	Environment	Main goals
Customers	Age: 18+ Education: Anything Computer skills: Normal	Usage: As needed Training: Not required % of user base: Around 80%	Technical environment: Laptop/PC/Tablet Real environment: Anywhere with an internet connection	<ul style="list-style-type: none"> • Check account • Service overview • Request service • Cancel service • Update user information
Companies	Age: 18+ Education: Anything Computer skills: Above average	Usage: As needed Training: Not required % of user base: Around 20%	Technical environment: Laptop/PC/Tablet Real environment: At work/home	<ul style="list-style-type: none"> • Check account • Service overview • Request service • Cancel service

Table 6: User groups and their details

5.2 Requirement list

This chapter will cover the requirements for the system, both functional and non-functional. These requirement were picked upon reviewing the project description, evaluation of other similar websites and from interviews. All requirements were given a priority on how essential they are to the system. The following table [Table 7] shows how high of a priority the requirements are.

Priorities	
A	Absolutely essential
B	Useful, not critical
C	Nice to have

Table 7: Priority definitions

5.2.1 Functional requirements

The following table [Table 8] shows the functional requirements the team decided upon and their priority.

Number	Description	Use Case Number(s)	Priority	Implemented
1	Users can create an account.	1	A	Done
2	Users can log in to the system.	2, 3, 4, 5, 6, 7	A	Done
3	Users can view an invoice.	2, 3, 4, 5	A	Done
4	Users can view their invoice history.	2, 3, 4, 5	A	Done
5	Users can search for invoice by date.	2, 3, 4, 5	A	Done
6	Users can view information about services and prices.	2, 6	A	Done

7	Users can order a subscription.	2, 7	A	Done
8	Users can cancel their subscriptions.	2, 7	A	Done
9	Users can get invoice sent by email.	2, 3, 4	B	Done
10	Users can request a new password.	2	B	Done
11	Users can request a transfer of subscriptions between addresses.	2	B	Done
12	Users can request a service for subscriptions.	2	B	Done
13	Users can request a service for disruptions.	2	B	Done
14	Users can print out an invoice.	2, 3, 4, 5	C	Done
15	Users can view an overview of their service history.	2	C	Done
16	Users can change their phone number.	2	C	Done
17	Users can request to change payer info for their account.	2	C	Done
18	Users can request to change their current payment method.	2	C	Done
19	Users can sign in with electronic ID.	2	C	Not done

Table 8: Functional requirements and their priorities

5.2.2 Non functional requirements

The following table [Table 9] shows the non functional requirements the team decided upon and their priority.

Number	Description	Priority
1	The system supports all main browsers	A
2	The system is easy to use and does not require training	A
3	The system is safe to use	A
4	The system has to adjust to the screen resolution.	A
5	Direct connection with network and monitoring equipment with SNMP.	C

Table 9: Non-functional requirements and their priorities

5.3 Evaluation of websites

It's important that the system is easy to use and has a good UX. By evaluating other similar web sites it's possible to get a grip on the systems layout and its functionality. Websites that were looked into included Norðurorka, Nova, Síminn and Vodafone. All of these websites have a very similar functionality and structure regarding their account information pages. Evaluating these websites and gathering their pros and cons helped a lot when the system prototypes were constructed. Overall they gave the team a great view on how the system could look like and function.

The team intends to take the best parts from each website, customize them to fit our system and implement them that way. The same goes for the bad parts, they will be talked about and the plan is to try to avoid them in the implementation while avoiding new ones. The system will most likely have some flaws and having evaluated other similar websites will hopefully make the team better at catching these flaws while in development. These results can be viewed in the following table [Table 10]

Website	Pros	Cons
Síminn	Login provides multiple ways to log in. Username / Email + password. Electronic identification (SIM or Debit / Credit card). SMS.	Very raw UI after login screen.
Nova	Login UI looks good. You can view all your invoices. You can view invoices by particular days. Easy to change your account information.	Very raw UI after login screen. Fetching account information is very slow. Login requires double authentication.
Vodafone	Nice UI when creating a new account. Simple and good UX when looking through internet usage.	Only one way to login. Homepage has a very raw UI. Only possible to fetch invoices from last six months.
Norðurorka	Simple UI. Easy to view invoices. Search by time period. Print out invoices possible. News and announcement.	Outdated UI.

Table 10: Evaluation of websites

5.4 Interviews

It was decided to conduct interviews with staff of Tengir early in the developing process to get a good sense of which parts of the system would be the most important ones and find out if any details were missing. The gathered information proved to be very helpful on improving the system analysis and their results can be view in the [results](#) chapter.

5.4.1 About the interviewees

Interviews were conducted with four people. All of them are currently employees at Tengir and all of them had some experience working with customers and their needs. Their age was in range of 30 - 50, three of them were female and one male. Three of them have a Bachelor's degree in Business Administration while one of them has a Diploma in Computer Science. One of the interviewee had more experience in helping customers working with

technical issues, such as TV disruptions or slow internet connection, while the others were mostly dealing with accounting issues and cancelling subscription.

5.4.2 Interpretation of interviews

The outcome of the interviews were evaluated and will be interpreted here. The questions that were asked can be found in [Appendix B](#).

All of the interviewees were currently working at Tengir hf. and were customers themselves. They all thought that the most important part as a customer was to be able to view their invoices, monthly and an overview for the whole year. Two of them thought it would be great to be able to cancel a subscription and change their account information, such as email address or telephone number.

5.4.2.1 Employees

As employees, when asked about how they would like to receive requests from customers they all wanted to receive an email about the requests. When asked about customers most frequent requests they all said that most of them wanted to view their invoices. One of them noted that she gets a lot of requests about canceling subscriptions and people complaining about buffering. One of the interviewee had a great suggestion that the system could show announcements on the front page for potential disruptions or issues. This was something that hadn't come up for the requirement list nor the backlog on Taiga.io so it was added to the requirement list as a B priority. Another one noted that if a company is paying for an employees invoices, that employee should not be able to view their invoices since they belong to the company, not the employee, also he would like to be able to download them as PDF.

5.4.2.2 Customers

As customers, interviewees thought it would be good to be able to see how much notice you have to give when cancelling a subscription. They also noted that customers often think they only have to cancel subscriptions at their telephone companies which they think automatically cancels their subscriptions with Tengir, which it does not.

5.5 Use cases

Use cases are a great way to get a better idea on how to implement requirements and also see how they affect other requirements. The following use cases goes through a few of the requirements that all have a big impact on the system and its design.

Name	User creates an account.
Number	1.
Priority	High.
Pre-condition	User is a customer of Tengir and has an email address in their database.
Description	User navigates to the website, goes to the service portal and registers for a new account.
Alternative Flow	User navigates to the website, goes to the service portal and tries to register for a new account. The email address is not registered in Tengir inner system and thus he is unable to register. He calls Tengir, registers an email with the staff and the staff puts the email into the database if he is in fact, a customer. Now that his email is in the database, the user tries to register for a new account and is now successful.
Post-condition	The user has created an account and can now log in.
Requirements	1
Actor(s)	Customer, businesses.

Table 11: Use case for user creating an account

Name	User logs in to the system
Number	2.
Priority	High.
Pre-condition	User already has an account.
Description	The user navigates to the website and logs in to the service portal.
Alternative Flow	User tries to log in to the service portal but the credentials are incorrect, he tries again and now successfully logs in.
Post-condition	The user is now logged in.
Requirements	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Actor(s)	Customer, businesses.

Table 12: Use case for user login

Name	User can view his invoices
Number	3.
Priority	High.
Pre-condition	User is logged in User has received at least one invoice.
Description	User logs in to the service portal. From the homepage the user will select the option to view his invoice.
Alternative Flow	User tries to log in to the service portal but the credentials are incorrect, he tries again and now successfully logs in. User has not received any invoices and a message will be displayed on the screen that to inform the user that he hasn't received a invoice.
Post-condition	The user is now logged in.
Requirements	2, 3, 4, 5, 9, 14
Actor(s)	Customer, businesses.

Table 13: Use case for user viewing his invoices

Name	Users can check invoice history
Number	4.
Priority	High.
Pre-condition	User is logged in. User has received at least one invoice.
Description	User logs in to the service portal. From the homepage the user navigates to his invoices and there he can see an overview of all his invoices.
Alternative Flow	User logs in to the service portal but hasn't received any invoices.
Post-condition	The user is now viewing an overview of his invoices.
Requirements	2, 3, 4, 5, 9, 14
Actor(s)	Customer, businesses.

Table 14: Use case for user checking his invoice history

Name	Users can search for invoices by dates.
Number	5.
Priority	High.
Pre-condition	User is logged in. User has at least one invoice.
Description	User logs in to the service portal. From the homepage the user navigates to his invoices, chooses his desired dates in the datepicker and searches.
Alternative Flow	User logs in to the service portal and searches by wrong dates, he changes the date to the right date and can now see all the invoices he wanted.
Post-condition	The user is now viewing his invoices by desired dates.
Requirements	2, 3, 4, 5
Actor(s)	Customer, businesses.

Table 15: Use case for user searching his invoices by date

Name	Users can view information about services and prices.
Number	6.
Priority	High.
Pre-condition	User is logged in.
Description	User logs in to the service portal. From the homepage the user will navigate to the services and prices category.
Alternative Flow	User tries to log in to the service portal but the credentials are incorrect, he tries again and now successfully logs in. He then navigates to the services and prices category.
Post-condition	The user is viewing an overview of the services and prices.
Requirements	2, 6
Actor(s)	Customer, businesses.

Table 16: Use case for user viewing information about services and prices

Name	User can order a subscription.
Number	7.
Priority	High.
Pre-condition	User is logged in. User has set up payment method.
Description	User logs in to the service portal. From the homepage the user will navigate to his subscriptions and can send in a request to order a new one.
Alternative Flow	User logs in to the service portal but hasn't set up payment method. He sets up a new payment method and can now order a new subscription.
Post-condition	The user has ordered a new subscription.
Requirements	2, 7, 8
Actor(s)	Customer, businesses.

Table 17: Use case for user order his subscription

5.6 Results

This chapter will cover the results of making the requirement analysis, what changed, what was added, what was removed and other information gathered.

The interviews turned out to be great and after evaluating them one of the A requirements was changed to a B requirement. There were also three new requirements added that hadn't been thought of which are shown in the following table [Table 18].

5.6.1 Requirements added

The following table [Table 18] shows the three new requirements which were added upon reviewing the interviews and feedback from them. It also shows whether they are functional or non functional.

Description	Priority	Requirement
If a company is paying the invoices for a person and registered on the website as a payment method, that person should not get an overview of the invoices history.	B	Non-functional
Users can see how much notice they need to give to cancel a subscription.	B	Functional
Users can download their invoices as a PDF	C	Functional

Table 18: New requirements from interviews

5.6.2 Requirements changed/removed

The table below [Table 19] shows what requirements changed or were simply removed during development. It also shows the reason why this action was taken.

Description of requirement	Action	Why
Users can view an overview of their subscription history.	Removed	The team initially thought that the subscription history was stored in the Inner system, but currently it only stores the current subscription.
Users can search for invoice by number.	Removed	During testing it came clear that this feature was not needed since most users didn't know the invoice number.
Users can change their email address.	Removed	Email address is used to link together users in the Inner system and our system..
Users can view an overview of their service history.	Changed B → C	When prototypes were showed to the staff and CEO it came clear that this feature was mostly nice to have and not fundamental.

Table 19: Changed requirements during development

6 System Design

This chapter will cover the design and structure of the system. This system is designed as a brand new service for customers of Tengir hf. to be used as a service portal. Much focus will be spent on making the UX as user friendly as possible to make sure the website will be practical and easy to use.

The intention is to go over the basic construction of the idea for the system which will be done by presenting various types of UML diagrams, such as navigation and flow diagrams. It will also include page prototypes to give a better feeling for the look of the system although they might vary a bit from the final version. Lastly it will cover the coding rules which will be maintained throughout the whole project for better understanding of the code for future developers of the system.

Following subchapters are a draft of the design and will likely go through a lot of change throughout the projects lifespan. They should be viewed as ideas and speculations about how the system could work and are useful for consulting stakeholders and professionals.

6.1 Page prototypes

Prototypes are one of the most important aspect on system designs. These prototypes help tremendously when designing the overall look of the website and can easily be changed later. The influence on these prototypes came mostly from the evaluation of similar websites in the [Requirement Analysis](#) chapter. These prototypes were also shown to the project owner and he was quite happy with how they turned out.

Do note that this will not be a complete set of all windows the website will include, although the most common actions, and a few more, will be showed. A short description for each prototype will be provided below it.

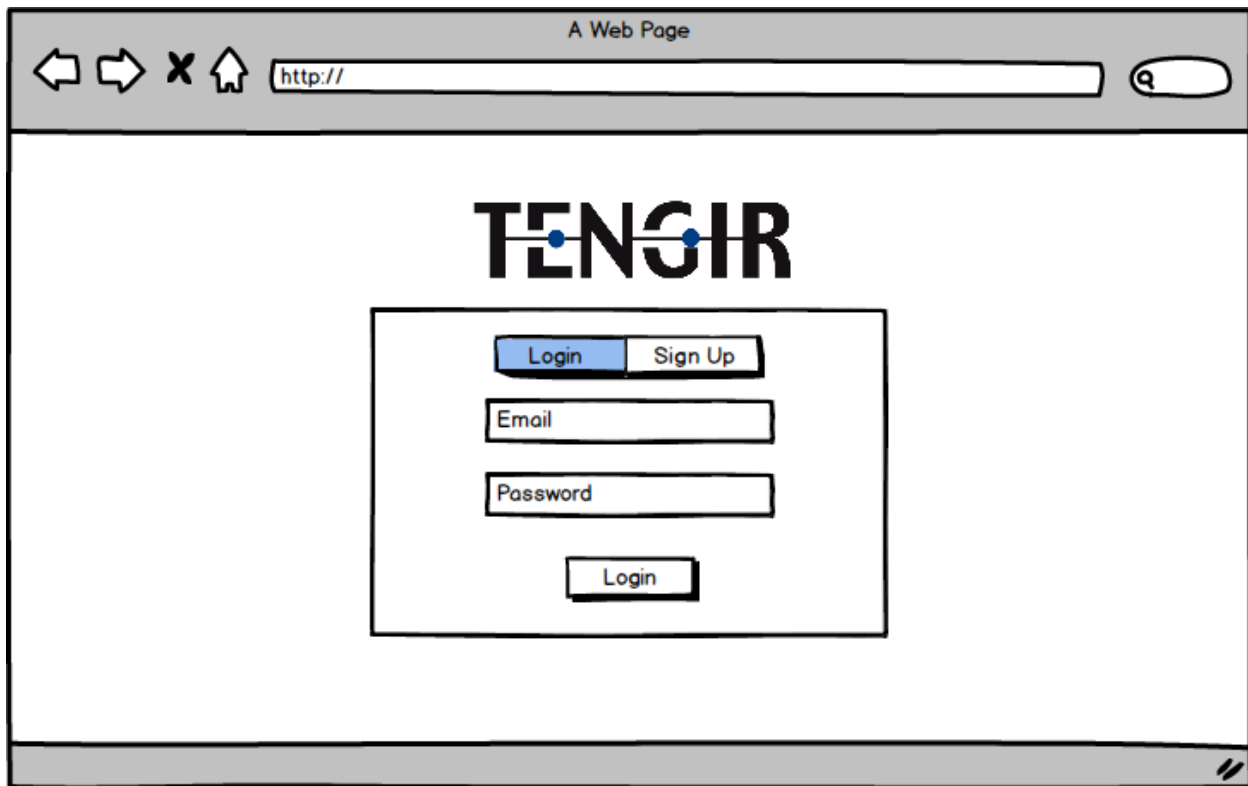


Image 1: Display of the user sign in page.

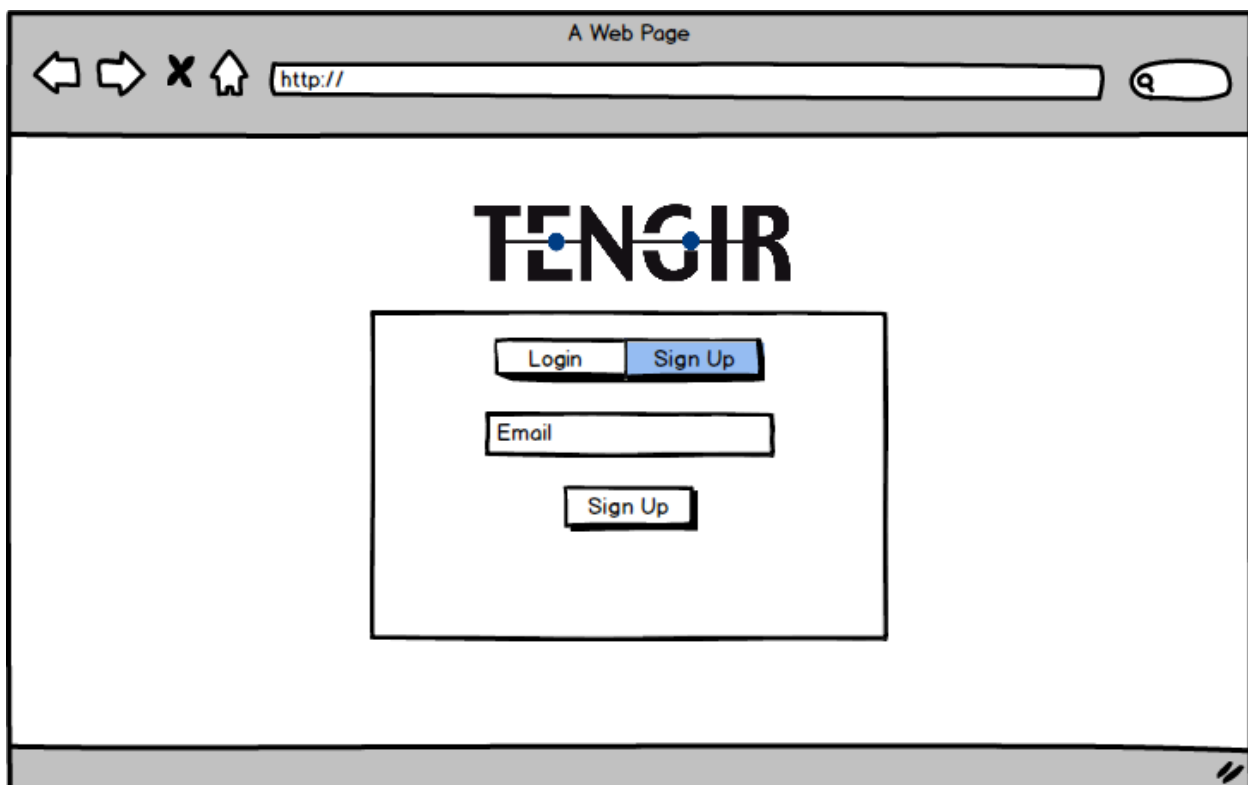


Image 2: Display of the user sign up page.



Image 3: Homepage of the system.

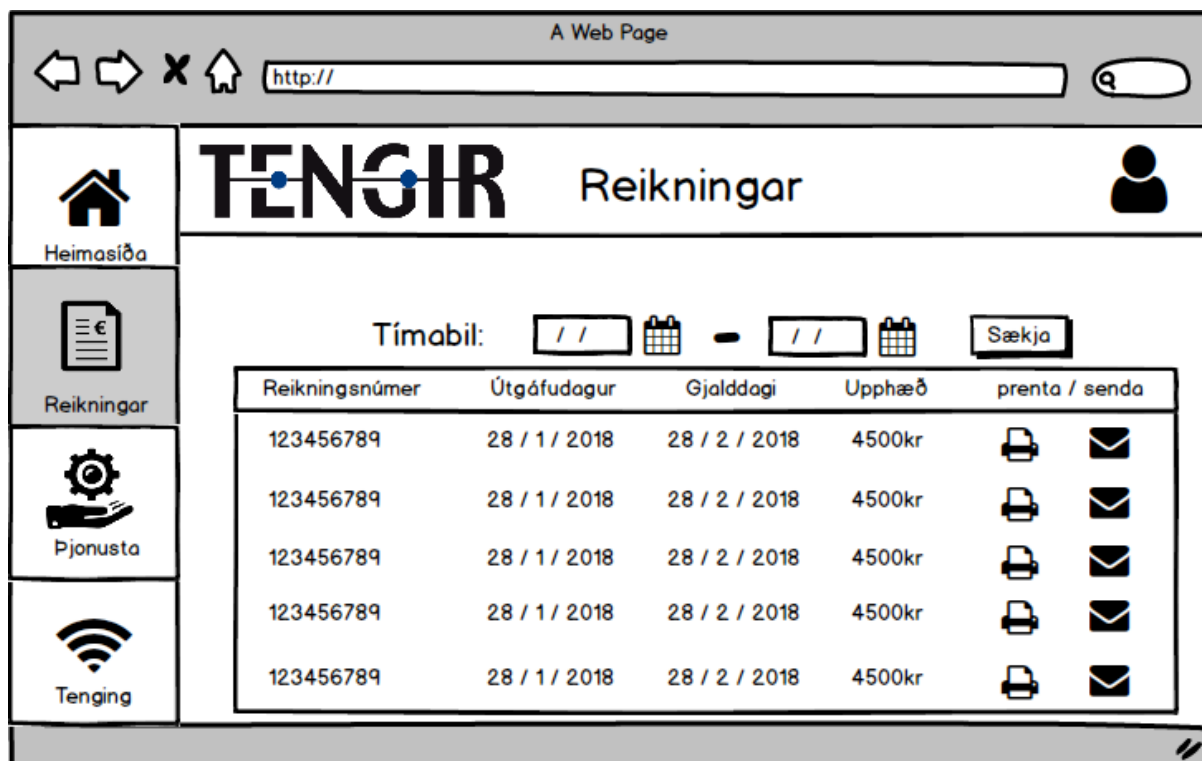


Image 4: Invoices for the current user.

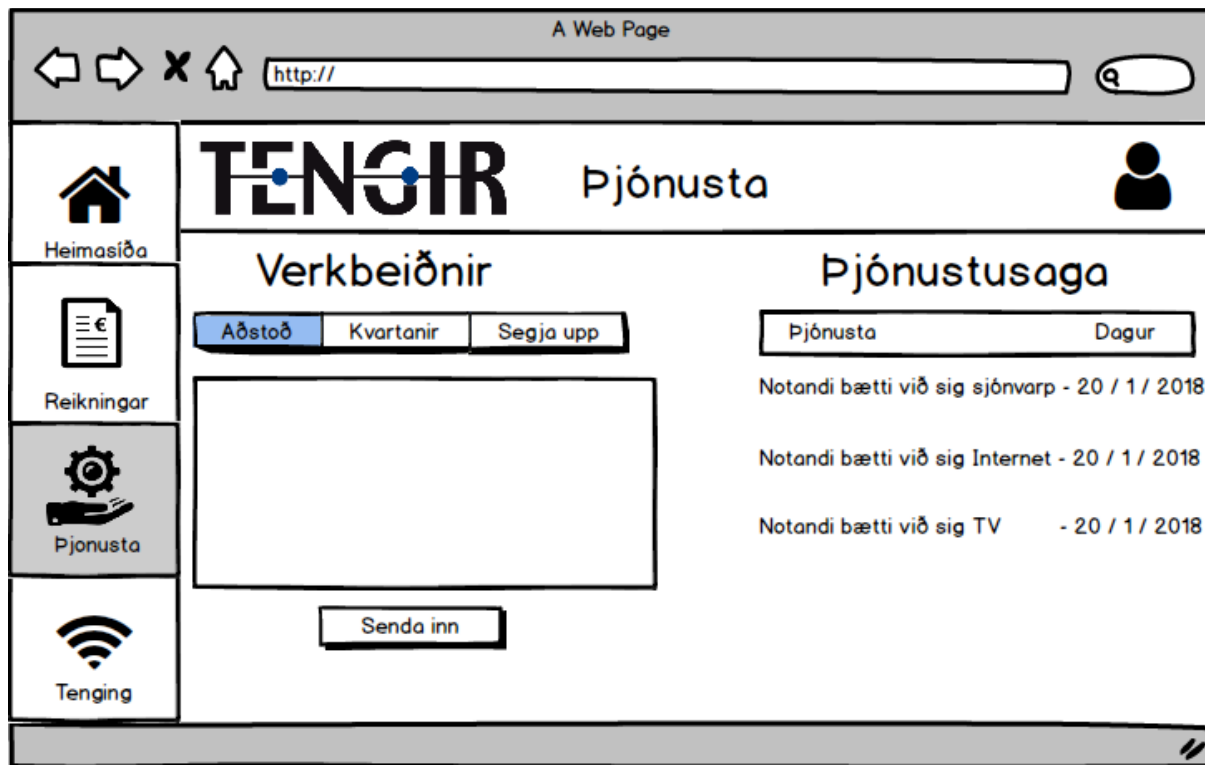


Image 5: Services that the user can request and service history.



Image 6: User current subscription information.

A Web Page

http://

Heimasíða

Reikningar

Þjónusta

Tenging

TENGIR Notenda stillingar

Upplýsingar - Jón Jónsson

Netfang Athugið að allar breytingar þurfa að vera samþykktar af starfsfólki Tengirs

Farsími

Heimasími

Kennitala greiðanda

Image 7: Account informations for the current user

6.2 Connection to other systems

Since the system heavily depends on other systems it was decided to create a simple image [Image 8] to show how they are connected. The service portal itself will mostly store data about its users and will receive invoices from Stólpi, the accounting system that Tengir uses to charge their customers. The second system that the service portal will rely on is the inner system that is developed by Tengir. That system is storing large amount of data about its users and their services, such as internet subscription, phone service and TV.

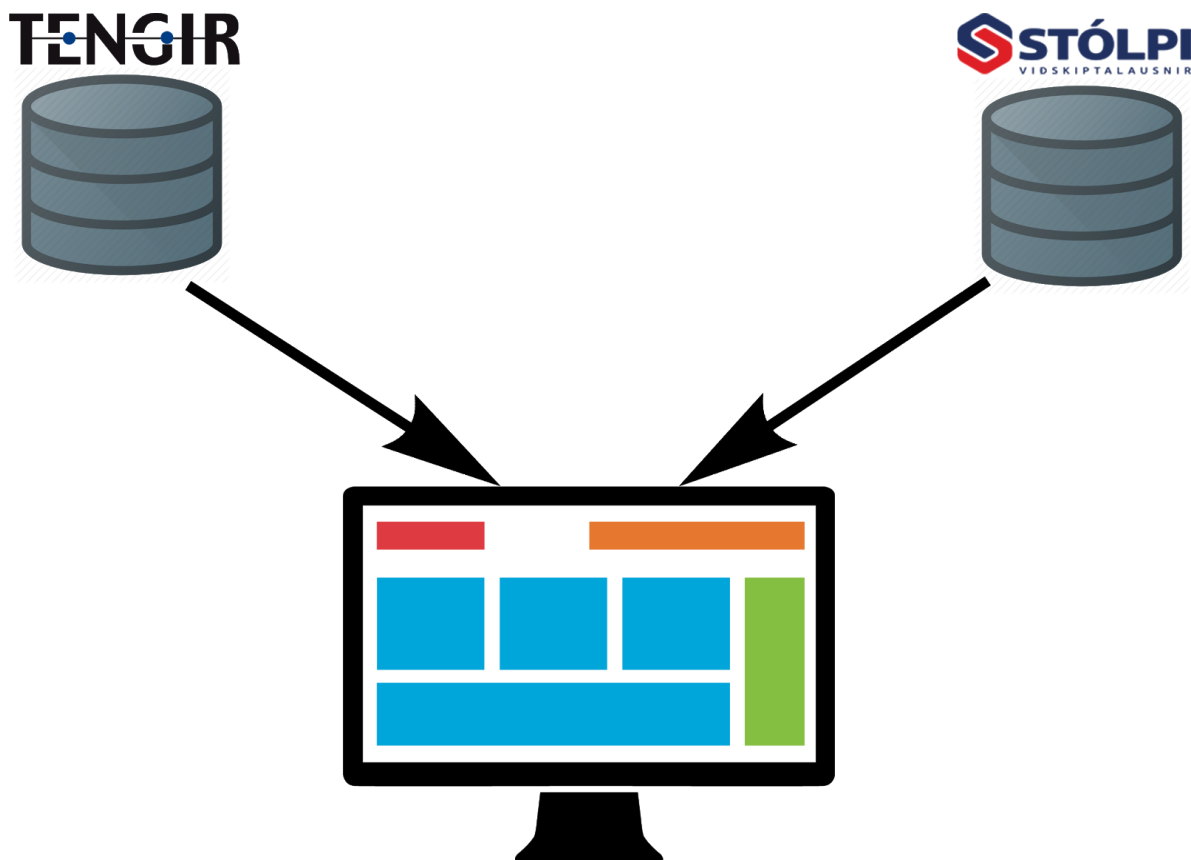


Image 8: Databases that the system depends on

6.3 Workflow

This chapter will go over a couple of diagrams the team made while designing the website. These diagrams are supposed to give the reader a better feel on how the user might use the website and are great for visualizing the flow of different tasks the user can do on the website. Due note, however, that in the final version of the system these diagrams might change, depending on how the final design will look and function.

6.3.1 Navigation diagrams

Navigation diagrams are great to show the flow of the website and how everything is linked together. Each box represent a webpage in the system and the arrows will show how the user can navigate from them. The system only has two user groups, as seen in the Requirement Analysis chapter, but the overall flow of the system will be the same. This is however, not a complete design since you can access most of the pages from the navigation bar on each page as seen in the prototypes earlier. It was decided to leave it out of the diagram to make it simpler to understand.

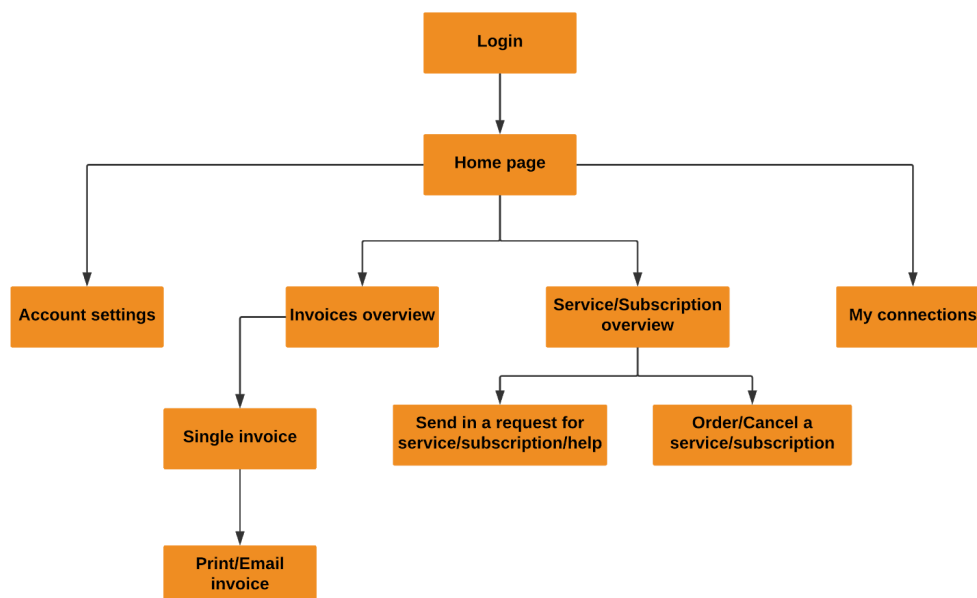


Image 9: Navigation diagram

6.3.2 Website flow

This chapter will go through a couple of diagrams that show the flow of some of the most important actions the customer might do while using the website. They will show which states the customer goes through, how they change and respond to various actions and in some cases how these actions affect the other systems the website counts on.

6.3.2.1 User settings

The following diagram [Image 10] shows which states the user goes through if changing his user settings. He tries to login to the website and if successful, is redirected to the homepage of the service. There he chooses the account settings, changes his desired settings and submits. The changes are sent to the Inner system for review, while they are up for review the user can still continue to use the website freely. If changes are valid they are approved, if they are not valid, they are disapproved.

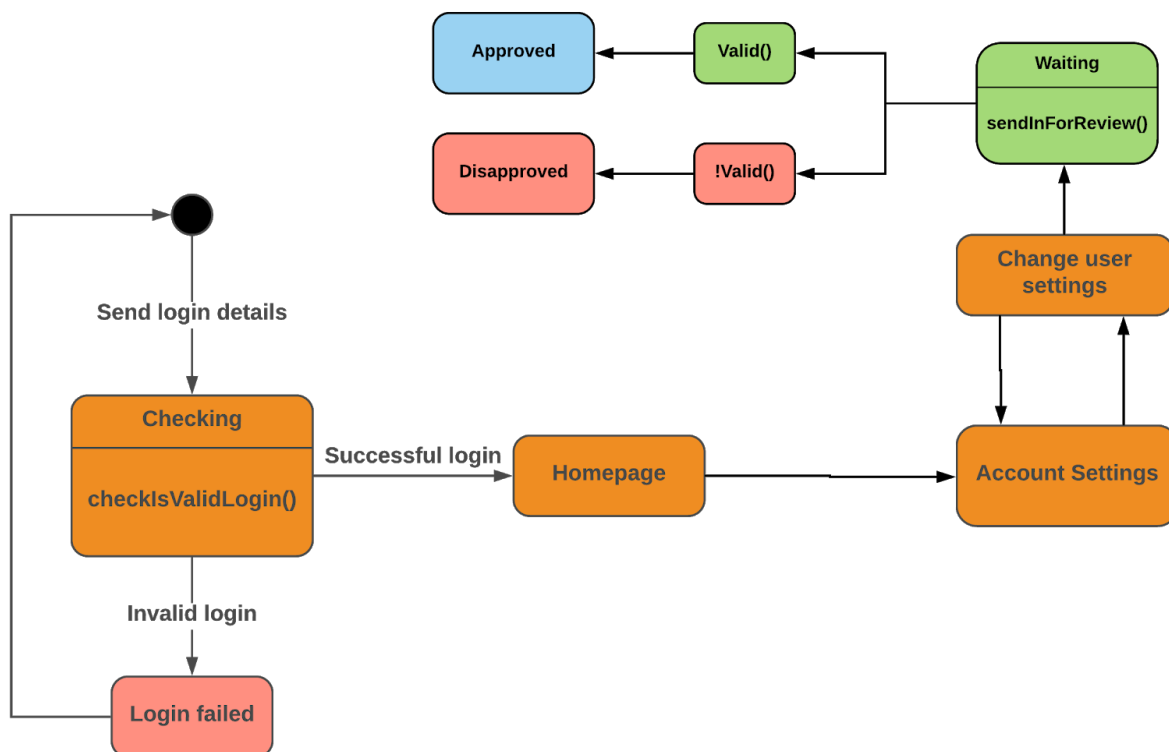


Image 10: Flow diagram

6.3.2.2 Printing / Emailing

This diagram [Image 11] shows the which states the user goes through when printing out or emailing a specific invoice of his choice. He tries to login to the website and if successful, is redirected to the homepage of the service. There he chooses the invoice overview page and finds his desired invoice. There he can select to either print it out or have it sent to his email account.

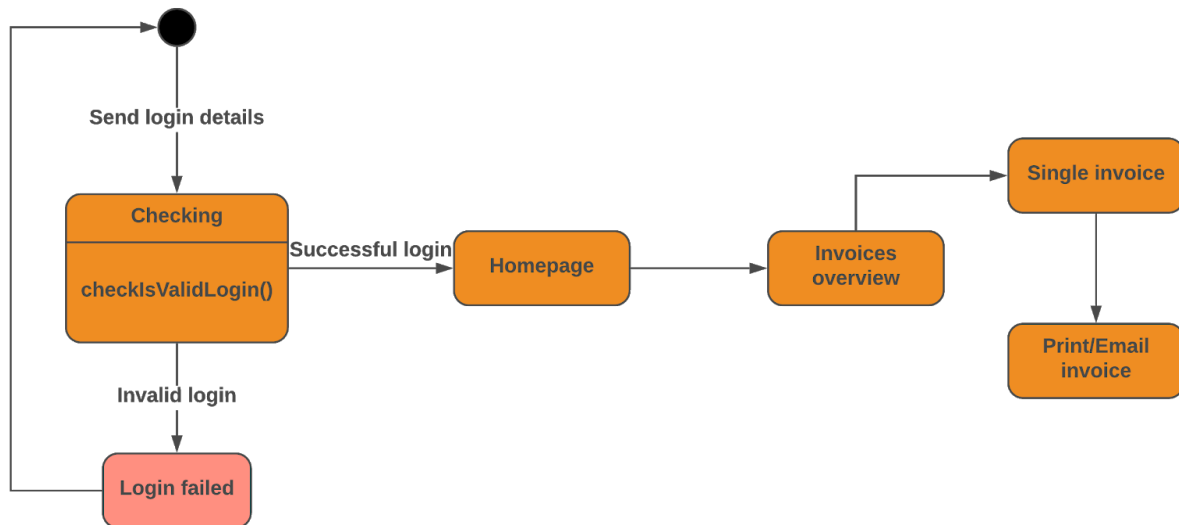


Image 11: Flow diagram

6.3.2.3 TV connection

The last diagram [Image 12] shows which states the user goes through when the user wants to add TV to his connections. He tries to login to the website and if successful, is redirected to the homepage of the service. There he chooses the services overview page and sends in his desired request. The request gets sent into the inner system and there it is up to staff of Tengir to approve and add the TV privileges.

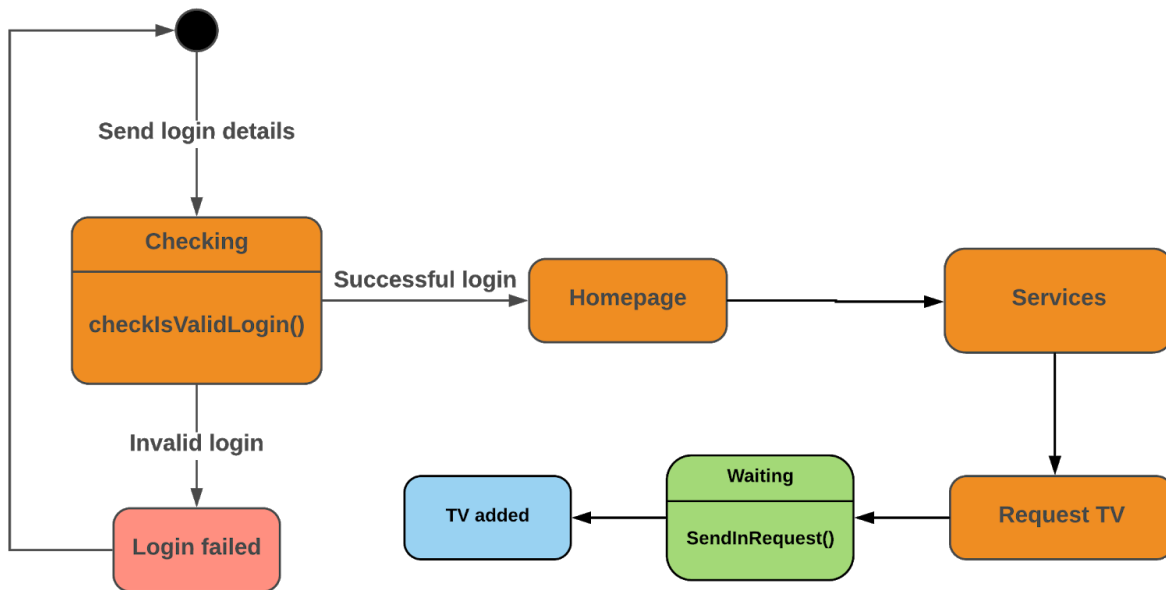


Image 12: Flow diagram

6.4 Coding rules

This will be a description of the coding conventions that will be used throughout this project. It was decided to follow a set of predetermined rules in order for the code to be consistent, improve readability and to make maintenance of the system easier in the future. The guidelines below will go through the general programming rules for C#, Javascript, HTML, CSS and the communication with the database. These rules should be followed in any aspect of the program.

6.4.1 General programming rules

- Our code will be written in English, including comments.
 - This is done in order to maintain consistency in the code. Other people are more likely to understand English rather than Icelandic.
- Comments for code functionality will be placed one line above it and with a single space separating the start of the comment and the comment itself.
 - For better explanation and understanding.

```
// Comment example  
public string customerName;
```
- Code lines are not allowed to be more than 100 symbols.
 - Makes the code more readable.
- The tab key will be used for indenting in all our code.
 - More consistent rather than using spaces.

6.4.2 C# / JavaScript code rules

Naming

- All names, for classes, variables and functions, shall be informative.
 - In order to make the code more readable and easier to understand.
- Variable names will use camelCasing.
 - `public string customerName;`
- Classes and functions will use PascalCasing
 - `public class UserInvoices;`
 - `public void GetUserInvoices;`

- Private variables will start with an underscore line.
 - `private int _db;`
- Constants and readonly variables will be in all uppercase letters, with an underscore to induce spacing to make the code more straightforward.
 - `public const int MAX_NUMBER = 5;`
 - `public readonly string SSN = "1111903059";`

Functions, if/else, loops and their declarations

- Curly brackets will be located below function declarations, if/else statements and loops. If there is only one condition in if/else/loop declarations, curly brackets may be omitted.

<pre>// Only one condition if(x == NULL) return 0; else return 1;</pre>	<pre>// Curly brackets located below function declaration public int f (int varName) { // DO STUFF // C# function return stuff; }</pre>	<pre>// Curly bracket located on same line public int f (int varName) { // DO STUFF // Javascript function return stuff; }</pre>
---	---	--

- The condition for if, else if and while loops shall be placed directly after the statement

<pre>// Correct if(i == 1) return 0;</pre>	<pre>//Not correct, single space between if and () if (i == 1) return 0;</pre>
--	--

- A single space will separate variables and operators.

<pre>// Correct for (int i = 0; i < 10; i++) doStuff();</pre>	<pre>// Not correct! for(int i=0;i<10;i++) doStuff();</pre>
--	--

6.4.3 HTML

- All tags will be properly closed, in lowercase and empty tags will have a trailing slash.
 - `<html> </html>`
 - `<input type="text" name="name" id="name" />`
- All tags must be properly nested, with one tab between.

```
<ul>
  <li>
    <p>Some text</p>
  </li>
</ul>
```

- All attributes will be enclosed in double quotes.

6.4.4 CSS

- Properties will be in lowercase and curly brackets will be located at the start of each property.

```
#example {
  background-color: gray;
  font-family: sans-serif;
}
```

- All Selectors will use camelCasing.
 - `#idSelector`
 - `.classSelector`

6.5 Database

The databases that the application needs to communicate with are different from each other. The Inner system at Tengir is implemented with a SQL server while Stólpi is a Microsoft Access database. The application database will be an SQL server that will store informations of the users and their notifications. Entities from the database will be instantiated with lazy loading to ensure that the program will run smoothly.

6.5.1 Rules

- Table, column and row names will be in English, and should always be explanatory for their functionality.
- All table properties will use camelCasing.
- All tables will include an ID, which will be the primary key for each element. The ID will be auto incremented, unique for all elements.

6.6 Testing

As always, testing is an important part in every single project. However with this project, where the system is almost entirely relying on fetching data from other systems, it was decided to only implement unit testing. The reason that the system is only allowed to fetch data and not change it was a request from Tengir. This makes sense since it would not be a good idea to allow a normal user to update important informations like the SSN for each invoice.

The built in MSTest was used for the unit testing, although both NUnit and XUnit were both looked at. It was decided to go for MSTest since the team had the most experience working with that framework. Most of the testing was done on the services and repositories. This was done with interfaces and dependency injection by creating a mock repositories that the services would rely on.

6.7 Websites style

Inspiration:

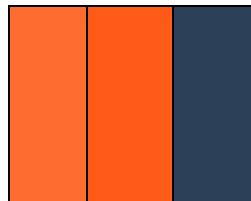
It was decided to look at [Tengirs official website](#) to draw inspiration when designing the style and look of the system. What was mostly looked into was the style and colors their website uses to find a theme to apply to this system.

Theme:

The main theme of their website is orange and white with blue colored buttons and it was decided to use the exact same colors to hold consistency between pages. It would not make sense to go from their website to this one if the theme wouldn't be the same, it would make the UX worse as the user might feel like he would have been navigated to an entire different website.

Colors:

These three colors in the following palette were mostly used, the two orange ones were used a lot on larger containers, like the navigation sidebar while the blue was used mostly for various buttons.



6.8 Solution

The solution will be made with a .NET Core 2.0 back end and JavaScript/CSS/HTML front end. In order for Tengir to be able to build on top of this project, the Areas feature will be used. This changes the hierarchy in such a way that all controllers and views for the user portal will be stored under an Area folder.

Services

All services will be stored in a Service folder along with their Interfaces. Those services will then be injected in the Startup script. Amongst those services is an Email sender, using an API from SendGrid, and a service for each database.

Connection strings

Sensitive information will be located in the secrets JSON file that will only be accessible by the server machine. This is information such as connection strings for databases and API keys for external applications.

Html to PDF

A package called Rotativa was used for the project to display HTML code as a PDF file. This is only used when users want to take a closer look at their invoice, potentially print it out or send it to their email account.

7 Progress overview

This section will go through the project releases and their focus and goals. It was decided to divide the project into multiple releases, where each release has a different goal. The time period of the releases was mostly decided on the due dates set by Reykjavík University. Following subchapters will briefly cover those releases but more informations about their sprints, retrospectives and burndown charts can be found in [Appendix A](#).

7.1 Organization Plan [24. Jan]

A lot of effort was put into organizing the project. Sprint Zero was solely focused on this release and successful at that. Tasks that were accomplished in this sprint included making the Work Schedule and the Organization planning chapter.

The following image [Image 13] shows the burndown chart for this release. From the graph, it is obvious that there came a couple of days where no progress was made. One of the main reasons for that is that this was the first week and the team hadn't come up with a schedule and something to enforce that some actual progress was being made. Later in this release the team had chosen a Scrum master and had made a work schedule and after that everything came together and the team finished this release strong.



Image 13: Burndown chart for first release.

7.2 Design and Analysis [7. Feb]

This release focused on the [Requirement Analysis](#), [System Design](#), updated [Work Schedule](#), [Risk Analysis](#) and the [Progress Overview](#). All these chapters were worked on during [Sprint 1](#) and were due before Wednesday the 7th of February. A lot more time was spent during this sprint than the initial one due to more workload and that resulted in a longer sprint.

The following image [Image 14] shows the burndown chart for this release. From the graph, it is again obvious that there were days where no progress was being made. This time it was due to heavy workload in other courses. The team members worked hard to finish the projects in other courses and managed to finish most of the tasks set for this release.

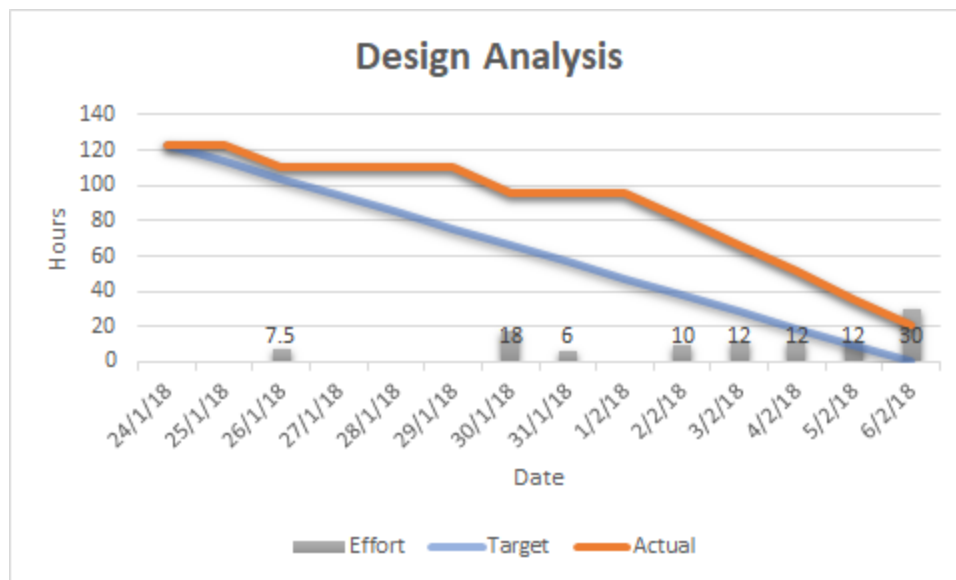


Image 14: Burndown chart for the second release

7.3 Prototype [11. Mar]

These two sprints, [Sprint 2](#) and [Sprint 3](#), focused mostly on setting the project up and starting some general programming work. Getting the project up on Github, getting connections to the other system it counts on to fetch data from and looking into different front end technologies. Programming work that were done included email verification and registration of a new user.

The following image [Image 15] shows the burndown chart for this release. The team followed a coordinated workflow and managed to stay pretty much on track. All main features were implemented and some extra ones as well, such as prototypes for views and extra services.

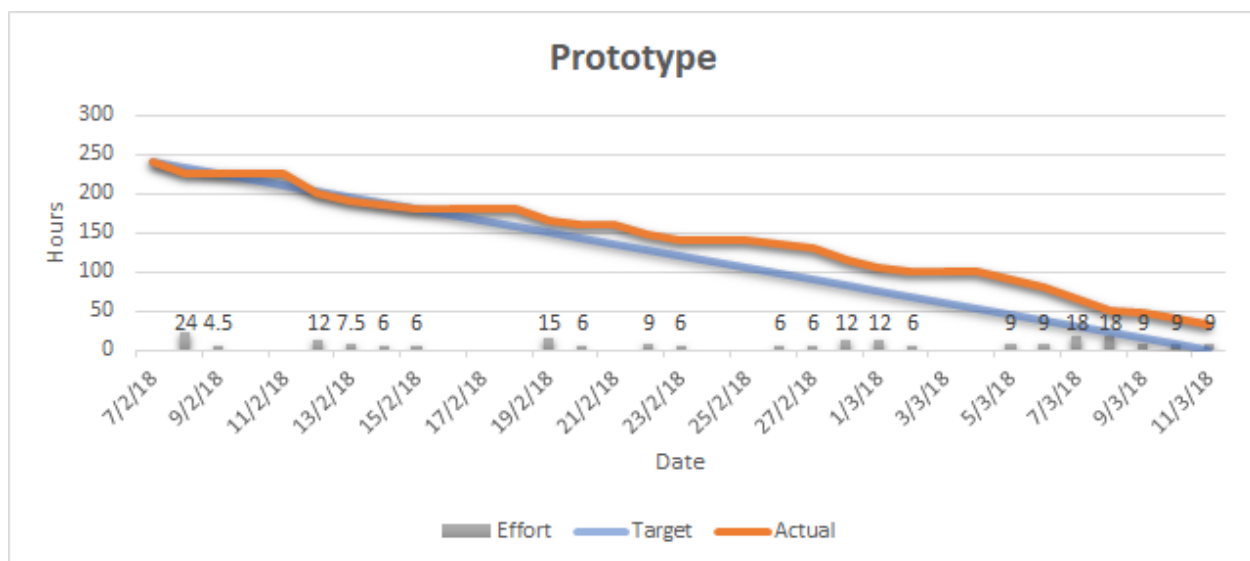


Image 15: Burndown chart for the third release

7.4 Final product [11. May]

This release contained five sprints and was the final release in this project. The release started with [sprint 4](#) and ended with [sprint 8](#). During this time there were a lot of diverse task that included a lot of programming work, documentation, meetings and more interviews with the staff at Tengir to name a few. Programming tasks included fetching Invoices from the Stólpi database, sending in service requests, printing and emailing invoices as PDF.

The following image [Image 16] shows the burndown chart for this release. As seen from the graph not much work was done during the exam period from 20. Mars - 15. April. This was however expected, but the team had hoped to have accomplished a little bit more but the overall workload at school was higher than expected. After the exam period the team could spend all their time on the project and a lot was accomplished during this time. The team followed a coordinated workflow and managed to stay pretty much on track. By splitting up this time period into multiple week long sprints, the team managed complete all functional requirements but one. By having such small and focused sprints proved to be incredibly good and the team was really happy with the outcome of this project.

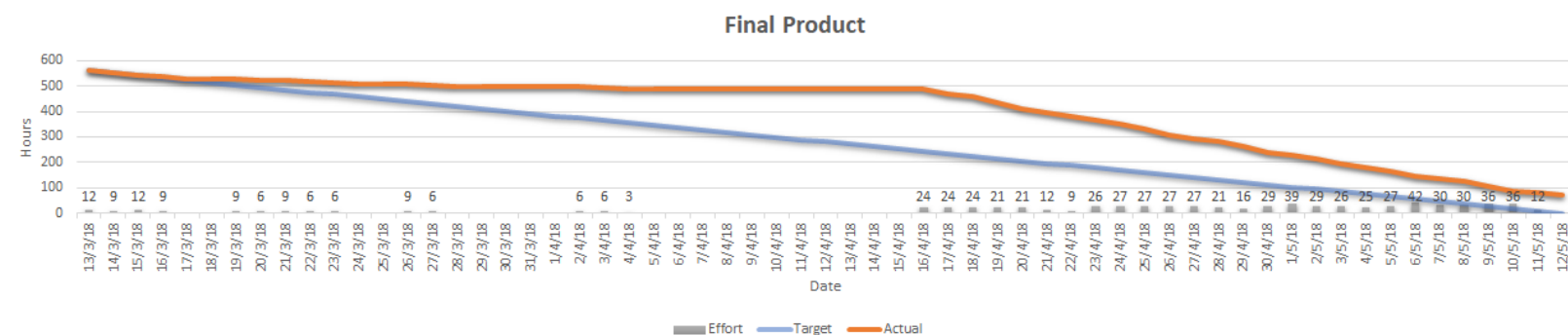


Image 16: Burndown chart for the third release

8 Project retrospective

This chapter will go through the retrospective for the whole project. Even though working on this project has been going great overall, there are always some things that could have gone better. Following subchapters will go through what went well, what went badly and what the team would do differently if the project was starting today. It will also cover possible future work of this project and also give a special thanks to people that helped during this time.

8.1 What went well

Throughout the projects' lifespan, the team used Taiga to set up the sprints, the backlog and to keep track of everything, although an excel sheet was used to create the burndown charts. Having everything set up to be able to look it up quickly was great and worked well.

The team was assembled of people who knew each other well, so communication was excellent. The whole team was based on Akureyri during development so meeting up and working together was easy. We had excellent facilities which Tengir provides us with and creating a good working environment was essential in delivering a good product and the team had fun working together.

8.2 What went badly

All team members were taking the same courses that semester, which included some very difficult courses. At times the workload got extremely heavy in these courses and thus time had to be taken from this project to be able to finish other courses projects before their deadlines. This resulted in the team having to spend countless evenings in the earlier weeks working late, sometimes to midnight, to be able to finish the sprints on a high note.

Although using Taiga for the sprints turned out great, using it to automatically create the burndown chart didn't work out like the team had hoped and as a result, had to be made in an excel sheet, manually.

8.3 What the team would do differently

There was a bit of a learning curve involved in using Taiga, it would have been better to have been able to utilize everything it offered, including the automatic burndown chart, so learning how it works sooner would be great.

Working on a solution to the Microsoft access database took the team up to a month, with breaks, and was the main problem hindering the team working on something else. It would have been ideal to begin sooner and seek help earlier from Ingvar to shorten the time it took.

Having a dedicated working schedule set up which the team would follow thoroughly would have made it easier to work, this was something that the team had planned and had set up in the beginning of the semester but unfortunately the workload got too intense at times which made it almost impossible to follow.

8.4 Future work

There are a lot of exciting things to build on this project including, SNMP connection, more Areas to add on, Sign in with an electronic ID, Integrating Tengir other systems onto this system and request forms.

8.4.1 SNMP connection

Direct connection with network and monitoring equipment with SNMP was a bonus requirement Tengir provided which unfortunately, the team didn't end up having the time to do which could be a cool feature, it would allow users to change various things about their ports themselves on the website.

8.4.2 Area addon

As of now the system has one Area, Users, which is thought of regarding normal users, e.g. people in households. Making a new Area for companies could be added onto it, it would not change the system that much, it would only give them a more thorough access to the system, maybe displaying more detailed versions of some features.

8.4.3 Electronic ID

Users are only able to sign in to the website using an email address and a password, a requirement which was added was signing in with an electronic ID, people are getting quite used to being able to do that nowadays, so adding that might be a rational decision.

8.4.4 Integrating other systems

Tengir mentioned they might want to make this project the base of all their projects. A big task to do would be to integrate their systems into this system. This is probably one of the biggest tasks left to do and one of the most essentials if their end goal is truly making it the base.

8.4.5 Request forms

As it is now, the user has to write every bit himself e.g. when requesting a service, or a complaint. What could be added are forms, the user would choose the type of request he is going to write and a corresponding forms would appear for him to fill in. Not only would this make it easier to get all points across from the user, it would also be very beneficial for the staff at Tengir as they would always get all the required information needed, thus no need for contacting the user about some piece of information missing.

8.5 Special thanks

The team would like to thank Tengir hf. for the projects' proposal and the facilities they provided. It was excellent and each team member even got an extra screen to work on which was great when it came to programming.

Special thanks will have to be given to Gunnar Björn, the project owner, Ingvar Karl and various staff at Tengir. They were very helpful during development and gave valuable feedback, help when asked and were more than happy to be interviewed during the making of the requirement analysis.

The team would also like to thank its instructor Hildi Andriksdóttir for her contribution to the project, the feedback she gave throughout the project was excellent and very helpful.

Appendix A

This appendix will cover all the sprints that were finished throughout the project. It will go through their description, retrospective and sprint tasks. Each sprint will be followed by a burndown chart representing the sprint. It will also include the interview questions that were asked during the interviews in sprint one.

A.1 Sprint Zero [18.jan - 24.jan]

This is the initial sprint. It is used to assemble the team and lay the lines. The expected result of this sprint is that the team is well prepared and can proceed to work systematically on tasks.

A.1.1 Retrospective

What worked well:

The teamwork started off great, every team member has registered and set up the technical environment and will be ready to take on the upcoming tasks. Most of the main tasks have been assigned to a sprint and a good overview has been established.

What could be improved:

[Taiga.io](#), the project management platform, is used for the project and should have been adapted earlier. It had a bit of a learning curve and bulked up activity was imported at a late stage in the sprint which resulted in the sprint tracking being inaccurate.

What will the team commit to in the next sprint:

The next sprint will focus mainly on theoretical aspect and design of the system. The team will get ready for the first release by making prototypes, requirement analysis and more. The team will also stick to Taiga and commit to fully learn how it works.

A.1.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Work schedule	2	Viðar	5,5
Organization planning	8	Arnar, Atli, Viðar	41
Sprint/backlog setup	2	Arnar, Atli	9

Table 20: Effort made on sprint zero

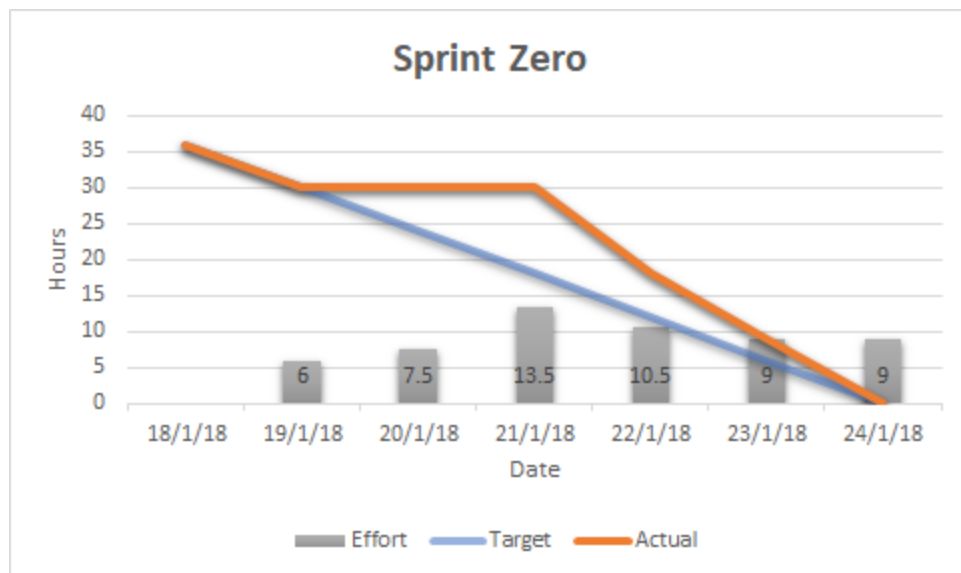


Image 17: Burndown for sprint zero.

A.2 Sprint 1 [25.jan - 7.feb]

No programming will be done during this sprint. Team members will focus on the theoretical aspect, analyzing risks and requirements as well as designing prototypes and diagrams. By the end of the sprint, programming tasks will be well defined and conflicts should stay minimal.

A.2.1 Retrospective

What worked well:

After adapting Taiga, it proved to be great to keep track of the process so far and became really well appreciated and fully utilized.

What could be improved:

Considering that this sprint was the first long period, a lot of tasks were planned. The big projects from other courses were not taken into account and that resulted in a minor setback for the team.

What will the team commit to in the next sprint:

The next sprint will mainly focus on setting up all the technical environment for the project itself, e.g. Visual Studio and Github. Documentation will be fixed using pointers from our first examiner meeting. The team will focus on planning better ahead and splitting the tasks up more efficiently with regards to other courses to avoid setbacks again.

A.2.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
State diagram	1	Arnar	1
Coding rules	2	Viðar	4
Prototypes	3	Viðar	6
Navigation diagram	1	Arnar	2
Table schema	1	Atli	2
Class diagram	2	Atli	5

Connection to other systems	2	Atli, Viðar	5
Work schedule	5	Arnar, Atli, Viðar	15
Risk analysis	5	Arnar, Viðar	14,5
Evaluation of websites	3	Arnar, Atli, Viðar	8
Requirement list	5	Arnar, Atli, Viðar	15
Use cases	3	Arnar, Viðar	8
Interviews (requirements)	3	Arnar, Viðar	8
Progress overview	5	Atli	14

Table 21: Effort made on sprint 1

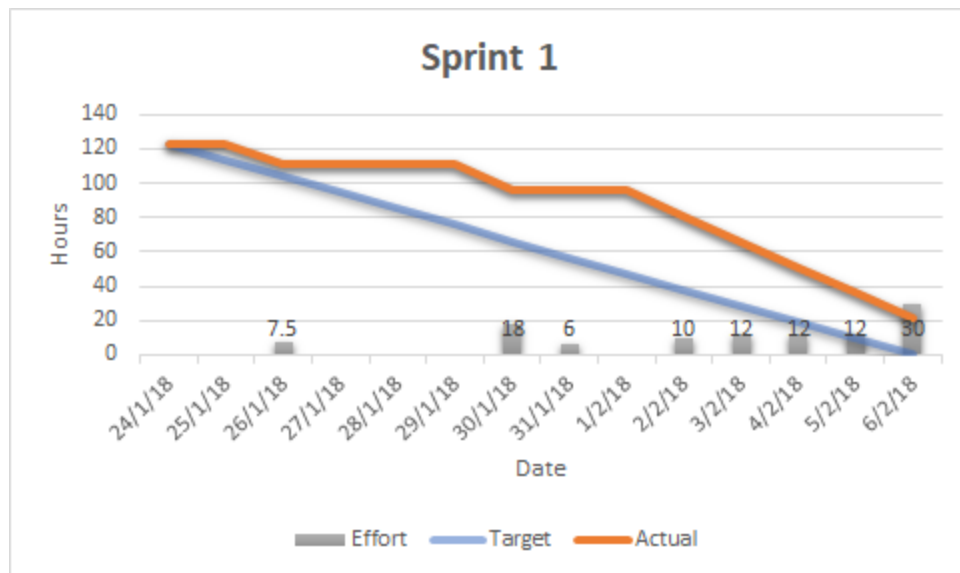


Image 18: Burndown for sprint 1.

A.3 Sprint 2 [8. feb - 25. feb]

Goal of this sprint was checking out possibilities of implementing the front-end of the system. Setting up the project on Github and establishing continuous integration. Refactoring the report, taking into account notes from Stefán which the team gathered during the first examiner meeting was also finished.

A.3.1 Retrospective

What worked well:

Refactoring the report went extremely well after receiving valuable feedback from Stefán. The team was also quite excited on setting up the project after working on numerous reports and a lot of researching was done for different kind of implementations.

What could be improved:

Some time went into the research and the team wanted to consider making the website a Single Page Application and thus React and Redux were looked into. However after consulting with Ingvar it was decided that it would be best not to try something completely new at this time. This resulted in the team wasting some time on something that was then scratched. This is something that the team could definitely improve and plan better

What will the team commit to in the next sprint:

Next sprint will go more into programming and setting up connections to the other systems that the application requires. It will also cover a lot of work on the final report and preparations for the second meeting with the instructors.

A.3.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Research front-ends	8	Arnar, Atli,	25
Set up project in IDE	4	Atli, Viðar	12
Github connection	5	Arnar, Atli	8
Set up CI	5	Arnar, Viðar	10
Fix documentation	10	Arnar, Atli, Viðar	29
Set up system database	5	Atli, Viðar	12

Table 22: Effort made on sprint 2

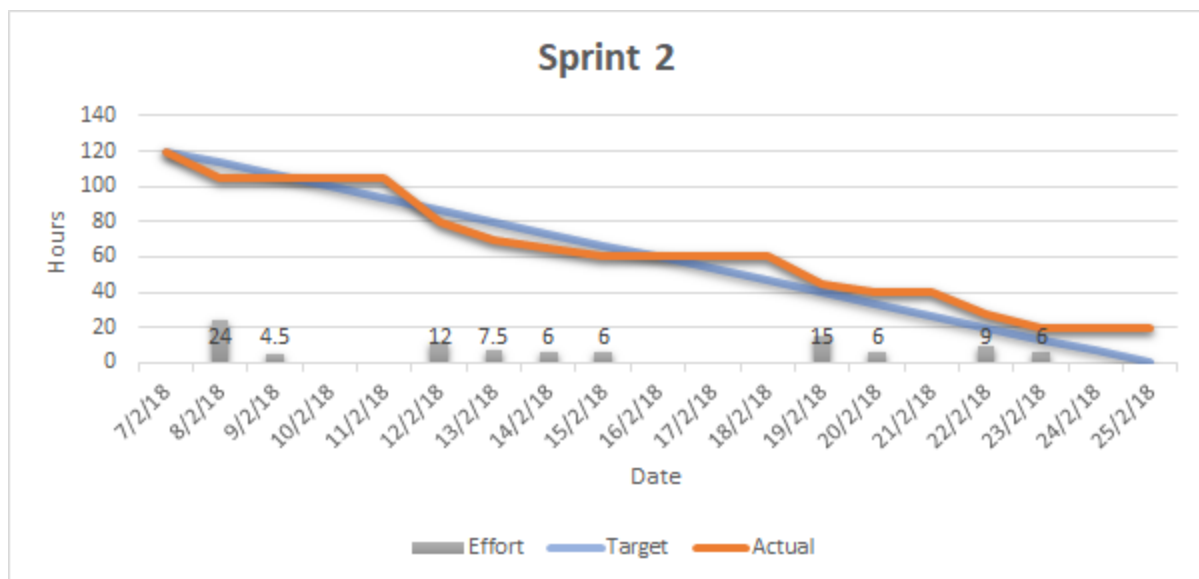


Image 19: Burndown chart for sprint 2

A.4 Sprint 3 [26. feb - 11. mar]

The main goal of this sprint was getting the connections to the other systems ready and starting general programming work on the project and getting the report ready for the next handin. The main programming tasks that were decided for this sprint were registration of new users given that the corresponding email address exists in the Inner database and their login after having verified their email after registration.

A.4.1 Retrospective

What worked well:

This is the first sprint that involves some programming work which the team was eager to finally start. There were a couple of projects due in other courses around this sprint so not a lot of big programming tasks were planned. The main goal was to get a connection to the Inner and Accounting system and to start learning our way around the project, e.g. getting familiar with the structure and everything. The team managed to plan well ahead and finish the projects in other courses as well as most the main goals of this sprint.

What could be improved:

Although most main tasks were completed, one was not. The connection to the Accounting system couldn't be established as the team ran into some issues regarding that. A solution will be worked on with Gunnar Björn and Ingvar in the next sprint.

What will the team commit to in the next sprint:

The next sprint will take time in the exam period with final projects in all courses and exams right after them so a not a lot of tasks will be planned. It will mostly focus on fixing the documentation, getting ready for the second examiner meeting and meetings with Ingvar. This sprint went rather well and the team will try to continue to work this way in the following sprints.

A.4.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Register account	5	Arnar, Viðar	15
Login after registration	5	Atli, Arnar	15
Connection with Inner system	12	Arnar, Atli, Viðar	36
Email Verification	8	Atli, Viðar	24
Fix documentation	10	Arnar, Atli, Viðar	33

Table 23: Effort made on sprint 3

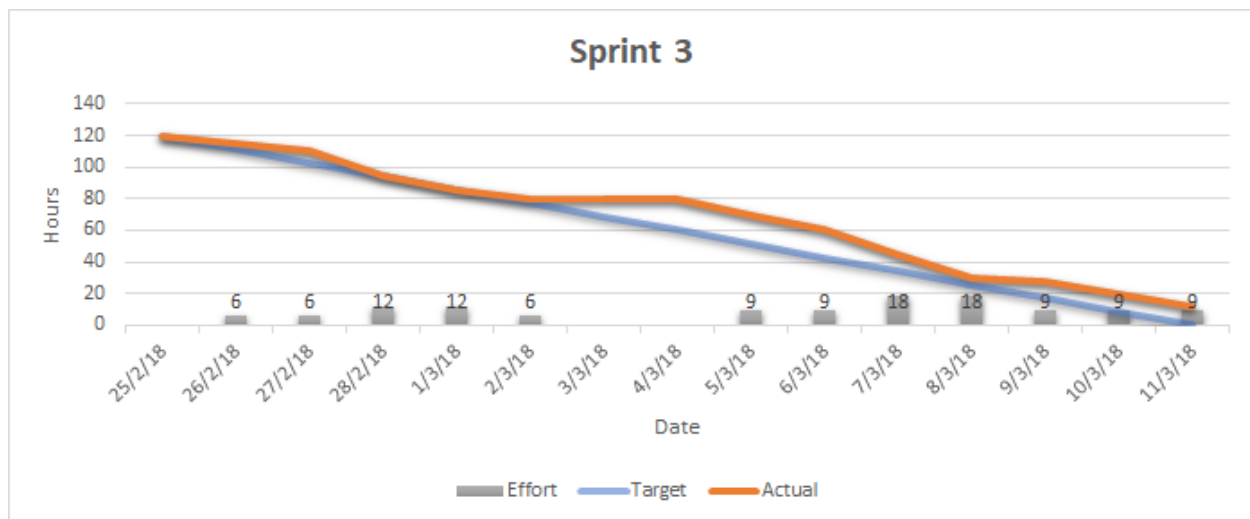


Image 20: Burndown chart for sprint 3

A.5 Sprint 4 [12. mar - 13. apr]

The exam period proved to be more difficult than predicted. This was due to three final projects in other courses that the team had not anticipated. Less hours were spent on this project and tasks handled were mostly meetings and documentation related. More effort was made on trying to connect to the accounting database.

A.5.1 Retrospective

What worked well

Tasks were limited and the team did a good job on deciding which tasks were important and short enough to be completed during the sprint.

What could be improved

The team could have realized sooner that the workload was so heavy and taken precautions that made the sprint more acceptable. Team members felt like they should be contributing more, although there was no time available.

What will the team commit to in the next sprint

Even though the courses have all ended, the team will take a better look before each sprint and see if anything might slow down the process and take precautions to minimize the damage.

A.5.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Presentation	9	Arnar, Atli, Viðar	27
Documentation	15	Arnar, Atli, Viðar	45
Programmer meeting	4	Arnar, Atli, Viðar	12
Database connection	8	Arnar, Atli, Viðar	24

Table 24: Effort made on sprint 4

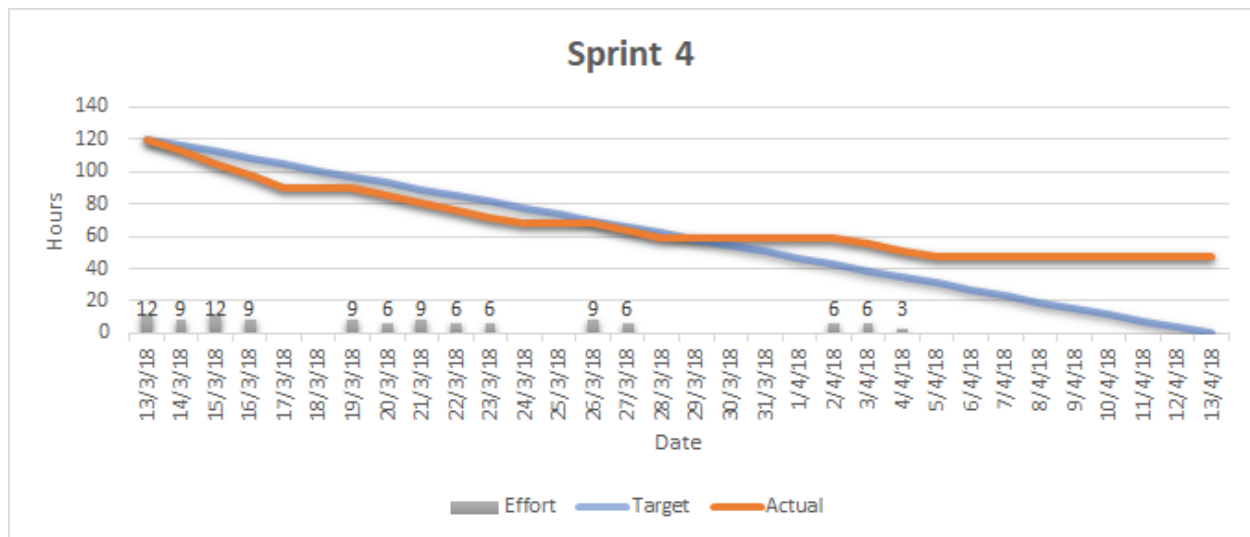


Image 21: Burndown chart for sprint 4

A.6 Sprint 5 [16. apr - 22. apr]

The main goal of this sprint was getting the connection to the Accounting system ready and start implementing features from the backlog. The main programming tasks that were decided for this sprint were log in of registered users, changing users account info and more general stuff, e.g. the invoices of the user and his connections.

A.6.1 Retrospective

What worked well:

This sprint was the first one where the team treated the project as a full-time job. The team was finished with other courses and exams so it could really dedicate all its time to the project. After getting the connection to the Accounting system ready things finally started to flow. It was decided to start implementing the functionality of the features only, and have only a very basic UI, for now. Each remaining tasks were sorted down to sprints and using Taiga to assign tasks to team members went well. Every team member got a task and a feature branch was made for each task. Treating it as a full-time job, working from 08:00 - 17:00 proved great and after a couple of setbacks, things finally started to click.

What could be improved:

After starting work on the features and the UI, the team quickly realized some pages had little information on them and might even be redundant. The team decided to get feedback from Gunnar, the Project Owner on what could be added. A decision was also made to get feedback from staff at Tengir on how the layout of the page could be.

What will the team commit to in the next sprint:

The team will continue to treat it as a full-time job, keeping the same work hours as in the previous sprint and put more effort into planning. Notes from Gunnar and the staff will be taken into account when designing the UI and added functionality. Tasks that were not finished here will get pushed over to the next one. The main programming tasks in the next sprint will be regarding user services, e.g. requesting and canceling a service.

A.6.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Register account	5	Arnar, Viðar	15
Database connection	8	Arnar, Atli, Viðar	40
Index View	3	Atli	12
Edit Account View	5	Atli	15
Invoice View	3	Arnar	9
Connection View	3	Arnar	9
Login	8	Viðar	24
Interview	3	Arnar, Atli, Viðar	5
Documentation	2	Arnar, Atli, Viðar	6

Table 25: Effort made on sprint 5

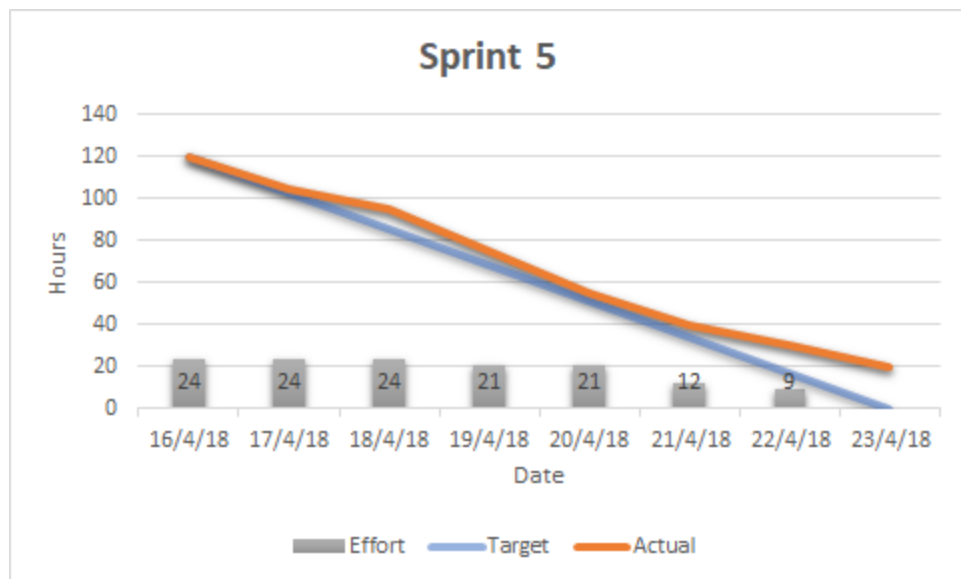


Image 22: Burndown chart for sprint 5

A.7 Sprint 6 [23. apr - 29. apr]

The main goal of this sprint was to keep the momentum from the last sprint going and continue implementing the features of the websites with minimum UI. The main programming tasks of this sprint were implementing the various service requests for the user, such as requesting a work order or sending in a complaint and being able to print out an invoice.

A.7.1 Retrospective

What worked well:

Things worked really well and the team managed to finish most of the features of the website that was set up with in the beginning of this project halfway through the sprint. A decision was made to move tasks from [sprint 7](#) over to this one in order to make room for team members that had no tasks.

What could be improved:

Tasks that the moved over only included improving the websites UI. The team had limited knowledge and skills in that frontend area, html / css / javascript, thus things slowly moved forward in the beginning of these tasks. Fortunately, since these tasks were from the next sprint after, the team wasn't that nervous about losing time, and after spending the first few days improving their skills in the frontend, things started to get better.

What will the team commit to in the next sprint:

The project is going great at this time and no major features are left to do. Major time will be spent on improving the websites UI and making it more responsive. The next sprint involves turning in this report before the third examiner meeting, thus some time will be spent working on that, including the User Manual and the Operational Manual.

A.7.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Users invoice activity	8	Arnar, Atli	36
Users service history	8	Arnar, Viðar	26
User orders a service	8	Atli, Viðar	26
User requests work order	5	Arnar, Atli	15
User can view his connections	8	Arnar, Viðar	24
User requests service	5	Atli, Viðar	16
User cancels service	3	Arnar, Atli	10
Documentation	5	Arnar, Atli, Viðar	18

Table 26: Effort made on sprint 6

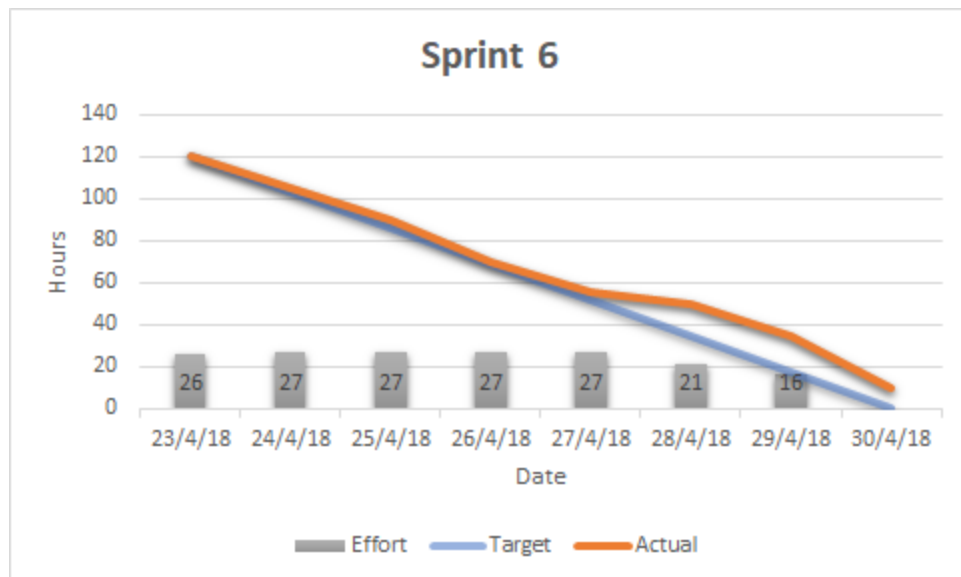


Image 23: Burndown chart for sprint 6

A.8 Sprint 7 [30. apr - 6. May]

The main goal of this sprint was to style the page and make it more responsive. Most major requirements had been finished here but some time was also spent on the smaller requirements. The main programming task in this sprint included sending an Invoice to a users email and fixing various bugs which came up when letting staff at Tengir try out the system. Meetings with Hildur and Ingvar were also held to get feedback and tips for the coming days. The last day of this sprint was used to prepare for the third examiner meeting.

A.8.1 Retrospective

What worked well:

Everything at this point was going great and according to plan. After this sprint all A and B requirements had been finished with only 1 C requirement left and the team was happy with what had been accomplished. Feedback from Ingvar and Hildur was taken into account when finalizing the requirements. Getting staff at Tengir to try out the system turned out great, the team got good suggestions on how to improve the system as well as finding bugs in the system.

What could be improved:

This sprint went by very quickly and the team might not have realized how closed it was to the final release and the last examiner meeting. With that in mind the team could have planned their time better as preparation for the examiner meeting was put on hold until the day before it.

What will the team commit to in the next sprint:

The next sprint is the final one, much effort will be spent on finalizing the look, making the UI better looking and in general making the website more user friendly. The team has planned a meeting with Ingvar to test out the system. The aim of that test is to give him a chance to break the system and find bugs in it for the team to fix.

A.8.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Email PDF	6	Atli, Viðar	22
Invoice UI	8	Arnar	32
Connection UI	4	Arnar	18
Homepage UI	4	Atli, Viðar	20
Service UI	4	Arnar	18
Settings UI	4	Atli	18
Meetings	1	Arnar, Atli, Viðar	12
Test runs with staff	3	Arnar, Atli, Viðar	12
Documentation	4	Arnar, Atli, Viðar	42
Prepare for examiner meeting	2	Arnar, Atli, Viðar	21

Table 27: Effort made on sprint 7

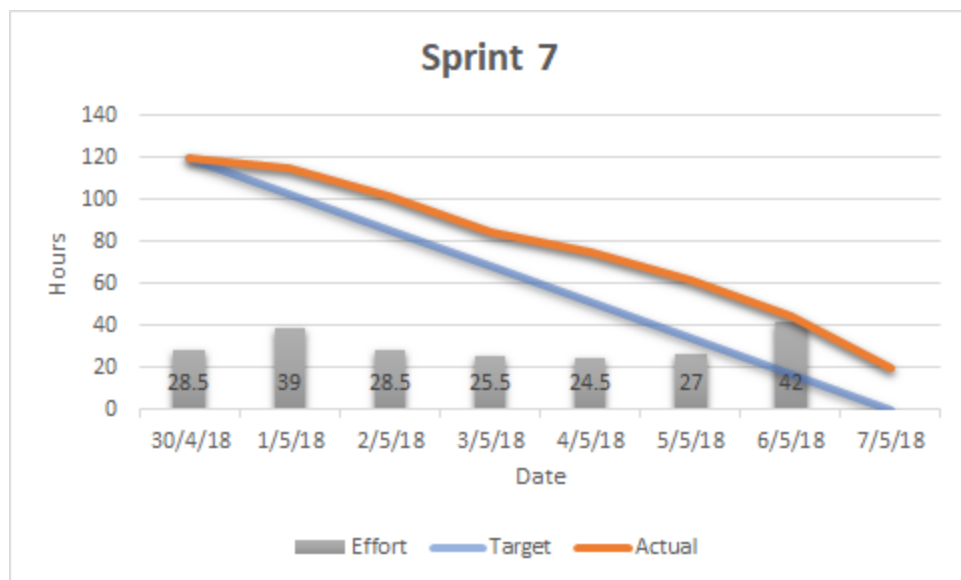


Image 24: Burndown chart for sprint 7

A.9 Sprint 8 [7. may - 14. May]

This was the final sprint for the project and the main focus of it was to improve the style of the website, fixing known bugs that had come up, setting up more unit tests, finishing up the documentation and finally making preparations for the public presentation. All but one requirements had been implemented and it was decided to leave it out and focus on improving other features. Two meetings were held during this sprint, one with the instructor, Hildur, while the other was held with Gunnar, the project owner.

The meeting with Hildur was mostly to ensure both documentation and the presentation were both close to being ready and also to show her the final features and the new style of the website. The other meeting with Gunnar was mostly on the same route as Hildur, that is, showing the new features and improved style.

Do note that it was decided to make this sprint a little bit longer than the other before in order to make the public presentation a part of it. The team will work on the presentation during the weekend (11. May - 13. May) but the work won't show up on the burndown chart. This is due to the fact that the report due date is on friday the 11th of May, three days before the public presentation. Another thing that will be worked on will be getting the project onto Tengir's server computer, to be able to show that it is online and in testing for staff at Tengir at the public presentation.

A.9.1 Retrospective

What worked well:

This sprint started with the last examiner meeting. The meeting went well, with a lot of great feedback along with some slightly negative comments, mostly about known issues. It was decided to fix these issues immediately to not forget anything. Letting Gunnar Björn and Hildur try out the system was great and ended up giving the team a couple of good suggestions on how to improve.

What could be improved:

There isn't much that could have been improved, the team had worked hard up to this point so there was no major task left which gave everyone a bit of breathing room. All in all, it was a great final sprint.

What will the team commit to in the next sprint:

As this is the last sprint the team will not commit to anything. Although, it is planned to help Ingvar set up the project on Tengir's server computer.

A.9.2 Sprint backlog & burndown chart

Description	Story point estimate	Developer	Hours spent
Style the page	10	Arnar, Atli, Viðar	60
Operations manual	5	Atli	12
Helpers on the page	5	Arnar	14
Final report	8	Arnar, Atli, Viðar	18
Unit Testing	5	Viðar	15
Presentation	5	Arnar, Atli, Viðar	15
Meetings	2	Arnar, Atli, Viðar	10

Table 28: Effort made on sprint 8

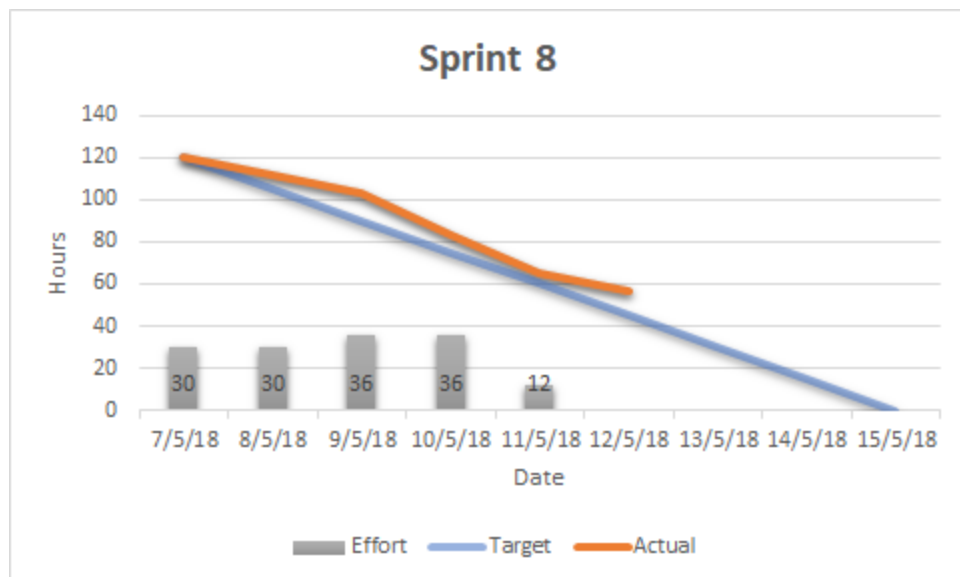


Image 25: Burndown chart for sprint 8

Appendix B

This appendix will cover the interview questions the team asked during the interviews in the [corresponding](#) chapter.

B.1 Interview questions

1. What would you like to be able to do and view on the website?
2. How would you like to receive requests from customers?
3. What are customers mostly asking for when contacting you?
4. Anything that you would like to add?

Appendix C

This appendix will cover some of abbreviations used in this report.

C.1 Glossary

The following table [Table 29] shows a few abbreviations used in this report and their meaning.

Word / Phrase	Meaning
CEO	Chief Executive Officer
UX	User Experience
UI	User Interface
CI	Continuous Integration
SNMP	Simple Network Management Protocol
SSN	Social Security Number (kennitala)
PDF	Portable Document Format
SIM	Subscriber Identity Module
SMS	Short Message Service
UML	Unified Modeling Language
RU	Reykjavík University

Table 29: Glossary

Appendix D

This appendix will cover additional reports that are included with the application.

D.1 Manuals

Manual for the application and future developers:

- [OperationalManual.pdf](#)