Final Project:
SmartGuide: Thor, the assistant of the future

Authors:
Einar Pórir Árnason
Magnús Torfi Magnússon

Project instructor:
Hildur Andrjesdóttir

May 17, 2019
# Contents

1 Introduction 4  
2 Project Description 4  
3 Networking 4  
4 Time Plan 6  
   4.1 Time Plan 6  
   4.2 Sprints 7  
      4.2.1 Sprint Plan 7  
5 Work Procedures 8  
   5.1 Work Methods 8  
   5.2 Personnel And Roles 9  
      5.2.1 Product Manager 9  
      5.2.2 Product Owner 9  
      5.2.3 Development Team 9  
   5.3 Workspace And Facilities 9  
   5.4 Technical Environment 9  
   5.5 Coding Conventions 11  
   5.6 Testing 11  
      5.6.1 Testing Platforms 11  
6 Product Backlog 11  
   6.1 User Types 12  
      6.1.1 Travelers 12  
      6.1.2 Guides 12  
   6.2 Story Points 12  
7 Risk Analysis 16  
8 Progress 18  
   8.1 Sprint 0 19  
   8.2 Sprint 1 20  
      8.2.1 Retrospective 20  
   8.3 Sprint 2 21  
      8.3.1 Retrospective 21
8.4 Sprint 3 ......................................................... 22
  8.4.1 Retrospective ........................................... 22
8.5 Sprint 4 ........................................................ 23
  8.5.1 Retrospective ........................................... 24
8.6 Sprint 5 ........................................................ 24
8.7 Sprint 6 ........................................................ 25
  8.7.1 Retrospective ........................................... 25
8.8 Sprint 7 ........................................................ 26
  8.8.1 Retrospective ........................................... 27
8.9 Sprint 8 ........................................................ 28
8.10 Sprint 8 ....................................................... 28
  8.10.1 Retrospective .......................................... 28

9 Project Retrospective 29
  9.1 What went well ........................................... 29
  9.2 What went badly ......................................... 29
  9.3 What would we do differently ........................... 29

A Chatbot Inputs And Flow 30
Abstract

This project aims to add an AR/VR extension to the SmartGuide app in the form of a chatbot. Users of the app are able to verbally interact with the chatbot and make a booking request for a trip with a tour guide.
1 Introduction

The SmartGuide app was first developed for companies to find available tour guides for hire in an easy and convenient way. Now it has also been opened up to individuals where they can get direct contact with tour guides, ask them questions and book a trip with them. The app has an easy to use search engine where you can search for what kind of activities you are interested in (e.g., hiking, sailing or a pub crawl), location and what language the guide must be able to speak. As of today the app is servicing Iceland and Norway but further expansions are in the works.

2 Project Description

The project aims at adding an extension to the SmartGuide app that enables virtual/augmented reality to the app. At first the idea is that private travelers can be answered by a bot when a guide is not available or if they ask common questions that can easily be answered in a standard way. The chatbot uses an AI that is already in place and is being trained to answer, by analysing communication between guides and other users in the app.

The user is guided through the booking form by a 3D avatar named Thor. After the user has been asked a series of question about the trip he/she is looking to book, the booking is sent on to a database with the information the user has specified to Thor. A unique bookingID is generated, so that multiple guides can message the user back, offering their services.

The next step in this development is location specific information through the chatbot. For example a tourist could view a location with his phone’s camera and summon the 3D avatar who would then give them relevant information about the location they are at.

3 Networking

As seen on figure 1, most of the network is handled with Amazon services. Guides and travellers authenticate through Amazon API Gateway and from there have access to the SmartGuide system. The AR/VR part of the system is run by Amazon applications.
Figure 1: Network Diagram
Figure 3, that we got from SmartGuide, shows how the systems are connected through the Amazon API Gateway.

4 Time Plan

4.1 Time Plan

<table>
<thead>
<tr>
<th>Period</th>
<th>Capacity per person</th>
<th>Total capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily</td>
<td>4 hours</td>
<td>8 hours</td>
</tr>
<tr>
<td>Weekly</td>
<td>5 days</td>
<td>10 days</td>
</tr>
<tr>
<td>Length</td>
<td>15 weeks</td>
<td>30 weeks</td>
</tr>
<tr>
<td>Total</td>
<td>300 hours</td>
<td>600 hours</td>
</tr>
</tbody>
</table>

Table 1: Estimated Capacity

As the table 1 shows, at least 20 hours will be dedicated each week per person to this project. The plan is to work Monday through Friday from 9:00
to 13:00 each of those days. Due to workload in other classes, it should be a realistic estimate. However, this is the minimum requirement and as the project comes along, more time may have to be dedicated to it.

4.2 Sprints

Tasks for each sprint will be planned at the start of that sprint. At the end of sprints a review will be held where team members and project owner will go over progress, what was good and what could have been done better, along with updating the requirements list.

4.2.1 Sprint Plan

Sprint 1 (feb 4. - feb 17.):
In this sprint the team has planned to work on progress report, risk analysis, initial system design and requirements analysis.

Sprint 2 (feb 18. - march 3.)
This sprint will be dedicated to studying AWS sumerian and ionic framework. The team will not start working through the functional tasks in Table 2, but instead build smaller apps as practise to see what is feasible, and also as a proof of concept of what can be done further down the road. This work will translate directly into less time needed to work on the actual app, as it will help identify problems on a much smaller scale than working on the SmartGuide app directly, making it much easier to analyze and deal with those problems before implementing features in the SmartGuide app.

Sprint 3 (march 4. - march 17.)
In this sprint the team will start implementing features for the project. A working chatbot demo to demonstrate how it will function and start work on implementing the AWS Sumerian scene and AWS Lex chatbot into an Ionic project.

Sprint 4 (march 18. - march 31.)
In this sprint the team will improve the chatbot demo to verify user input, and start working on opening a AWS Sumerian VR scene in an Ionic app.
Sprint 5 (April 1. - April 14.)
In this sprint team members have their exams, so no tasks or stories will be planned in this sprint.

Sprint 6 (April 15. - April 28.)
In this sprint the team will continue to improve the Lex bot and the Sumerian scene. The bot needs to be able to respond to more utterances and therefore the AWS Lambda code needs to be updated to verify more user inputs. The testing of the AWS Lambda code will also start. Work on activating the VR scene in an Ionic app will continue.

Sprint 7 (April 29. - May 12.)
In this sprint the team will have to make a Cordova plug-in to communicate with native Android apps. The AWS Lambda code for AWS Lex with connections to the SmartGuide API will be completed. As this sprint comes along more tasks will be added to it as needed. Since there has been some trouble with foreseeing what problems might arise with this new technology being used in this project, with extra work hours spent on research that comes with it, the method settled on was updating the sprint as tasks are completed.

Sprint 8 (May 13. - May 17.)
In this sprint the team will put the final touches on the project and finish what tasks are able to be completed.

5 Work Procedures

5.1 Work Methods
For this project the team has decided to adapt as much of Scrum methodology as is feasible for a two person group, since SmartGuide uses Scrum. Scrum is a part of the Agile movement and emphasizes decision making from real world results rather than speculation. There will not be a Scrum master role assigned.
5.2 Personnel And Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Manager</td>
<td>Haukur Viðar Jónsson</td>
</tr>
<tr>
<td>Product Owner</td>
<td>Ægir Finnson</td>
</tr>
<tr>
<td>Development Team</td>
<td>Einar Þórir Árnason</td>
</tr>
<tr>
<td>Development Team</td>
<td>Magnús Torfi Magnússon</td>
</tr>
</tbody>
</table>

5.2.1 Product Manager

The product manager is the CEO of SmartGuide. His responsibilities are similar to the product owner’s except for the technical aspect of the project. The product manager is a lot more involved than a typical stakeholder as defined in scrum, so it seems this role is more appropriate.

5.2.2 Product Owner

The product owner is the CTO of SmartGuide. His role is defining the product backlog and optimizing the product value, as well as offering the development team technical assistance.

5.2.3 Development Team

The development team works on developing the product.

5.3 Workspace And Facilities

SmartGuide has provided team members with an office in Glerárgata 34, Akureyri. There will be weekly meetings with Ægir Finnson (product owner) and constant communication with the SmartGuide team through Slack.

5.4 Technical Environment

The SmartGuide app is being built in the Ionic framework (HTML, CSS, Angular, Typescript), AWS Lex to make the chatbot, AWS lambda for the logic behind the chatbot and AWS sumerian for the AR/VR part of it. For collaboration we’ll be using github.
<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overleaf</td>
<td>Used to create and maintain reports</td>
</tr>
<tr>
<td>Google Chrome</td>
<td>Used when working with Amazon Web Service</td>
</tr>
<tr>
<td>Visual Studio Code</td>
<td>IDE for working with the ionic framework</td>
</tr>
<tr>
<td>Draw.io</td>
<td>Used to create diagrams</td>
</tr>
<tr>
<td>Google docs</td>
<td>Spreadsheet used to maintain a log of work hours</td>
</tr>
<tr>
<td>PowerPoint</td>
<td>Used to create slide shows for presentations</td>
</tr>
<tr>
<td>Trello.com</td>
<td>Used to organize workflow with a ticket system</td>
</tr>
</tbody>
</table>

The following picture shows how the software used to create the product works together. Ionic Cordova app starts the Sumerian AR/VR scene with the help of Amazon amplify. The verbal inputs from the user are processed with AWS Lex, and to respond back Lex calls AWS Polly, which is a text-to-speech service. The business logic behind the chatbot is run by AWS Lambda when Lex is prompted. AWS Lambda also runs the code is also responsible for inserting the order request in DynamoDB database.
5.5 Coding Conventions

The team will follow, the ES6 standard. To make it easier to follow the team has decided to use ESLint to make sure the standard is being followed. There is however an exception to this when it comes to AWS lambda as it demands that the require-style module (const moduleA = require('moduleA');) to be used.

5.6 Testing

Tests were implemented for the code that AWS Lambda execute for AWS Lex, as well as unit tests for the Ionic app. However a large portion of our code is executed by AWS Sumerian in the VR scene, and as of now, there is no testing platform for that code. So to verify that the code was working as intended, we relied heavily on console logging to figure out issues with the code in the VR scene. Even though this is not ideal, it is the best practice that is currently available to test the code in question.

5.6.1 Testing Platforms

**Jest:** Used to test code executed by AWS Lambda  
**Jasmine:** Used to test the Ionic app.

6 Product Backlog

To create a product backlog the team had a meeting with the product owner where the scope of this project was discussed. That meeting as well, as a brainstorming session between team members, produced several tasks for this project. These tasks are categorized by type, priority and if they are complete or incomplete.  
In Table 2, the broken down results from the analysis are listed.
6.1 User Types

6.1.1 Travelers

Private travelers that are able to connect with guides or a chatbot through the SmartGuide system.

6.1.2 Guides

Guides are professional tour guides that private travelers can contact through the SmartGuide system and ask for information and book tours with.

6.2 Story Points

To evaluate how complex the user stories are relative to one another the team estimated how many story points each is. A story that has four story points is four times bigger than a story that has one story point.

Table 2: Product backlog

<table>
<thead>
<tr>
<th>No</th>
<th>Description</th>
<th>Type</th>
<th>Priority</th>
<th>Status</th>
<th>Story point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create a functional api diagram</td>
<td>Report</td>
<td>High</td>
<td>Complete</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Create a network diagram</td>
<td>Report</td>
<td>High</td>
<td>Complete</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Traveler is asked by 3D avatar what places he would like to visit</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Create a script to generate sorted latex table for user stories</td>
<td>Report</td>
<td>High</td>
<td>Complete</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>The chatbot needs to be able to catch intents* from audio input and generate a response accordingly</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Requirement</td>
<td>Type</td>
<td>Priority</td>
<td>Completion</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------------------------------------------</td>
<td>------------</td>
<td>----------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Traveler is asked by 3D the avatar what type of activity / guide type he needs</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Traveler needs to be able to manually input an indicator that the microphone is active and the avatar is listening</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>The 3D avatar needs to wear a white shirt with the company logo</td>
<td>Non-functional</td>
<td>High</td>
<td>Complete</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>The chatbot can respond to pre defined phrases</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>Traveler can ask for a list of available guide types while in the 3D scene</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>Connect the chatbot to smartguide api where it gets its responses from</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>The android app needs to ask the user for permission to use the microphone and camera</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>A VR camera rig can be activated within the sumerian scene</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>Make a demo of the chatbot in a 3D scene</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>4</td>
</tr>
<tr>
<td>No.</td>
<td>Task Description</td>
<td>Functionality</td>
<td>Priority</td>
<td>Status</td>
<td>Weight</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>----------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>15</td>
<td>The chatbot verifies user input and asks user to repeat the phrase if it doesn’t understand it</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>Learn and implement a way to create and install a custom cordova plugin into the project for native ios/android functionality</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>5</td>
</tr>
<tr>
<td>17</td>
<td>Integrate sumerian into ionic</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>6</td>
</tr>
<tr>
<td>18</td>
<td>Traveler is asked when he’d like to go on the tour</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>The user needs to be able to view 3D avatar in AR mode in android app</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>7</td>
</tr>
<tr>
<td>20</td>
<td>The user needs to be able to view 3D scene in VR mode in android app</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>7</td>
</tr>
<tr>
<td>21</td>
<td>The 3D avatar needs to sound cheerful</td>
<td>Non-functional</td>
<td>Non-functional</td>
<td>Complete</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>The 3D avatar needs to perform relevant gestures</td>
<td>Non-functional</td>
<td>High</td>
<td>Complete</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>A guide can message traveler back using the bookingID attached to the message from the traveler</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Requirement</td>
<td>Functional Level</td>
<td>Priority</td>
<td>Completion</td>
<td>Notes</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>24</td>
<td>Traveler needs to see if the chatbot is listening to them while in the 3D scene</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>2</td>
</tr>
<tr>
<td>25</td>
<td>The skybox in the sumerian scene needs to be replaced by the mobile device camera in AR mode</td>
<td>Functional</td>
<td>High</td>
<td>Complete</td>
<td>6</td>
</tr>
<tr>
<td>26</td>
<td>Add a button to enable AR mode</td>
<td>Functional</td>
<td>Medium</td>
<td>Complete</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>Traveler can send guide(s) messages attached to their booking</td>
<td>Functional</td>
<td>High</td>
<td>Incomplete</td>
<td>3</td>
</tr>
<tr>
<td>28</td>
<td>Traveler can see a list of available guides while in the 3D scene</td>
<td>Functional</td>
<td>High</td>
<td>Incomplete</td>
<td>4</td>
</tr>
<tr>
<td>29</td>
<td>A traveler needs to be able to get information from a chatbot if a guide is not available</td>
<td>Functional</td>
<td>Medium</td>
<td>Incomplete</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td>A guide can get messages from traveler about bookings from bookingID</td>
<td>Functional</td>
<td>High</td>
<td>Incomplete</td>
<td>3</td>
</tr>
<tr>
<td>31</td>
<td>Traveler can book available guides</td>
<td>Functional</td>
<td>High</td>
<td>Incomplete</td>
<td>3</td>
</tr>
<tr>
<td>32</td>
<td>Traveler can open chat dialog with an aws lex bot</td>
<td>Functional</td>
<td>Medium</td>
<td>Incomplete</td>
<td>3</td>
</tr>
<tr>
<td>33</td>
<td>Build a text only view for the chatbot inside the application</td>
<td>Functional</td>
<td>Medium</td>
<td>Incomplete</td>
<td>4</td>
</tr>
</tbody>
</table>
**3D avatar needs to find an optimal ground to stand on when AR mode is activated**

**Functional**

**Medium**

**Incomplete**

**4**

**Traveler can get information about their current location from chatbot**

**Functional**

**Medium**

**Incomplete**

**8**

<table>
<thead>
<tr>
<th>No</th>
<th>Description</th>
<th>Severity</th>
<th>Probability</th>
<th>Priority</th>
<th>Mitigation</th>
<th>Response</th>
<th>Handler</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>3D avatar needs to find an optimal ground to stand on when AR mode is activated</td>
<td>Functional</td>
<td>Medium</td>
<td>Incomplete</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Traveler can get information about their current location from chatbot</td>
<td>Functional</td>
<td>Medium</td>
<td>Incomplete</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total: 114

*Intent – An intent represents an action that the user wants to perform. The user create a bot to support one or more related intents. For example, the user might create a bot that orders pizza and drinks.

7 Risk Analysis

Every project has some risks and to prepare for them the team did an risk analysis table. It was evaluated on a scale from one to five how severe(severity in table) the problem would be if it came up, and then how probable (probability in table) it is to occur. Then priority was assigned by multiplying thses two factors together (higher number = higher priority). These are both project specific risks, such as Amazon Web Service goes down, and more general risks such as a Team member becomes ill. Each risk factor has an assigned handler that is responsible to deal with the problem should it occur, as well as thought out mitigation and response to said problem. In table 3, the risk factors for this project are listed.

Table 3: Risk Analysis

<table>
<thead>
<tr>
<th>No</th>
<th>Description</th>
<th>Severity</th>
<th>Probability</th>
<th>Priority</th>
<th>Mitigation</th>
<th>Response</th>
<th>Handler</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Issue Description</td>
<td>Difficulty</td>
<td>Impact</td>
<td>Solution</td>
<td>Progress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------------------------------------------</td>
<td>------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Learning how to use AWS Sumerian is harder than initially planned.</td>
<td>3</td>
<td>3</td>
<td>Dedicate plenty of time to it</td>
<td>Magnús</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Learning how to use ionic framework is harder than initially planned.</td>
<td>3</td>
<td>3</td>
<td>Dedicate extra time to it</td>
<td>Einar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Amazon Web Services go down</td>
<td>5</td>
<td>1</td>
<td>N/A</td>
<td>Magnús</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mock API/Wait for Amazon Web Services to come back up</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Loss of data</td>
<td>5</td>
<td>1</td>
<td>Have backup of all relevant data(Git for code, cloudservices for documentation)</td>
<td>Einar</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fetch backup of relevant data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Workload from other classes</td>
<td>2</td>
<td>1</td>
<td>Stay up to date with other classes</td>
<td>Magnús</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Be flexible in time schedule should the need arise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date Range</td>
<td>Name</td>
<td>Issue Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>--------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. march - 18. march</td>
<td>Magnús</td>
<td>Large assignment in other class</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29. march - 30. march</td>
<td>Magnús</td>
<td>Large assignment in other class</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25. february - 4. march</td>
<td>Einar</td>
<td>Abnormally high workload in other courses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. march - 11. march</td>
<td>Einar</td>
<td>Abnormally high workload in other courses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29. march - 31. march</td>
<td>Einar</td>
<td>Abnormally high workload in other courses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. april - 16. april</td>
<td>Einar</td>
<td>Exam period</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. april - 16. april</td>
<td>Magnús</td>
<td>Exam period</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issue</th>
<th>Team member</th>
<th>Issue Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Team member’s computer breaks down.</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Lack of sufficient skill to complete certain functionalities</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>Team member or team members children come down with illness.</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date Range</th>
<th>Name</th>
<th>Issue Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18. feb - 25. feb</td>
<td>Magnús</td>
<td>Came down with an illness</td>
</tr>
<tr>
<td>26. feb - 27. feb</td>
<td>Magnús</td>
<td>Arm injury</td>
</tr>
<tr>
<td>3. may - 4. may</td>
<td>Magnús</td>
<td>Came down with an illness</td>
</tr>
</tbody>
</table>

## 8 Progress

The pie chart on figure 2 shows the work hours team members have spent on this project and how the time is distributed as of May 17, 2019.
The scale of the product was somewhat unknown in the beginning. As the project progressed, the user stories increased and became more clearly defined. To begin with, it took some time defining the project.

8.1 Sprint 0

In this sprint the team had their first meeting with the CEO and CTO of SmartGuide, on the 29. of January. Before that meeting the team had only a vague idea of what this project consisted of, but gained a much clearer vision of the project after that meeting. After that the team started planning work procedures.
8.2 Sprint 1

In this sprint the team started working on the progress report, risk analysis, system design and requirement analysis. A lot of progress was made with these tasks as can be seen in this report. All of these tasks are however alive throughout the project so they can be updated and altered, and will not be completely done until the project ends.

8.2.1 Retrospective

**What Went Well During The Sprint Cycle?**
The team worked well together. Being a two man development team has its advantages as all logistics become easier and more flexible.

**What Went Wrong During The Sprint Cycle?**
Nothing really went wrong but it would have improved efficiency to get access to the team’s work space earlier.

**What Could Be Done Differently To Improve?**

![Figure 3: Sprint 1 Burndown](image-url)
As the team only gained access to the workspace 4 days ago, it is hard to determine how the work process should be improved since the team hasn’t gotten into the routine of it yet. This will be addressed at the end of next sprint.

8.3 Sprint 2

Figure 4: Sprint 2 Burndown

In this sprint the team decided to dedicate time to study the tools that will be used in this project, AWS Sumerian, AWS lex and Ionic framework.

8.3.1 Retrospective

What Went Well During The Sprint Cycle?
The team gained experience and clarity with the Amazon tools and the technical capabilities of AR and VR in mobile devices
What Went Wrong During The Sprint Cycle?
Magnús came down with an illness and then had some injuries to his right arm so a lot of work hours were lost there.

What Could Be Done Differently To Improve?
These issues are not something you can avoid.

8.4 Sprint 3

This sprint went into creating a Sumerian scene and a bot to interact with. Then finding an optimal way to implement displaying the Sumerian scene in an Ionic project.

8.4.1 Retrospective

What Went Well During The Sprint Cycle?
A working demo app displaying an interactive Sumerian scene was created
in the end of the sprint with help of SmartGuide CTO Ægir Finnson.  

**What Went Wrong During The Sprint Cycle?**

Too much time was spent on building an implementation using a native Android project when there was already an Ionic support using Amazon Amplify. Also, it turns out only a small number of devices are supported by ARCore from Google. Magnús just got a new smartphone supported by ARCore, but Einar still has an unsupported smartphone.

**What Could Be Done Differently To Improve?**

More time should have been spent on research as some guides are harder to find. Get a device for Einar that supports ARCore.

### 8.5 Sprint 4

![Sprint 4 Burndown](image)

Figure 6: Sprint 4 Burndown

In this sprint the team made a working demo of the chatbot in a sumerian VR scene, continued working on the ionic application and had their second status presentation.
8.5.1 Retrospective

What Went Well During The Sprint Cycle?
The team got more experience working with the tools for this project. A working demo of the chatbot in a sumerian scene with validation of user inputs was produced, which was a significant milestone.

What Went Wrong During The Sprint Cycle?
A lot of time went into working on assignments for other classes, and as a result of that few work sessions were cancelled.

What Could Be Done Differently To Improve?
The team should have made a better job of estimating their workload from other classes and planned accordingly.

8.6 Sprint 5

Figure 7: Sprint 5 Burndown

This sprint was unusual as the team had their exam period at the same time. A conventional retrospective would not be applicable as this sprint was not...
planned and most of the team’s time went to studying for the exams.

8.7  Sprint 6

In this sprint the team continued trying to implement a solution to be able to open the VR scene in an Ionic app, as well as continue to expand the chatbot, as well as refactoring the code in AWS Lambda that controls the business logic behind it and writing tests for that code.

8.7.1  Retrospective

What Went Well During The Sprint Cycle?
Developing time has increased. Bugs were fixed and enabling VR in the app inside Chrome browser was achieved.

What Went Wrong During The Sprint Cycle?
Some components turned out to be much harder to implement than anticipated. For example, practically no solutions exist for activating native AR and VR functionality in Android for hybrid frameworks like Ionic. Writing tests for the code used in AWS lambda has proved a lot harder than anticipated. A lot of time was planned for development has gone into research that. As a lot of the tools used in this project are new to the team members, planning has become quite difficult as estimating how long a task will take has turned out to be almost complete guesswork. No effective solution has been found to this problem.

**What Could Be Done Differently To Improve?**
Developing a custom Cordova plugin for the app to activate AR and VR functionality in mobile devices is in consideration.

### 8.8 Sprint 7

![Sprint 7 Burndown](image)

**Figure 9: Sprint 7 Burndown**

This sprint went into tackling the harder problems, like native Android VR and AR mode and AWS Lambda functions improvements, since the team had
more time each day to focus on them. Early on, it was clear that the tasks were much more difficult than anticipated. So the tasks were re-evaluated and more time was estimated to complete them. The third status presentation was also in this sprint.

8.8.1 Retrospective

What Went Well During The Sprint Cycle?
The team accomplished a lot during this sprint. Major improvements were made to both the VR scene and the chatbot.

What Went Wrong During The Sprint Cycle?
Nothing went wrong during this sprint, but there were some long workdays that started to affect the efficiency of the team as this sprint came closer to ending.

What Could Be Done Differently To Improve?
Organizational improvements could be made. As has been for the whole period of the project, estimating how big tasks actually are and how long they will take to complete has proven difficult.
8.9  Sprint 8

8.10  Sprint 8

This sprint was unique as it only contained the last four and half days of the project time. The team made an assessment on what tasks could realistically be completed before hand-in, and what tasks should remain for future development.

8.10.1  Retrospective

The team worked hard on successfully completing as many tasks as possible before hand-in. As this is a really open project that is going to be developed further after hand-in, a lot of tasks and ideas were always going to be left outstanding. As this sprint was short, and completely focused on completion before hand-in, the team decided that the conventional retrospective of "what went well", "what went wrong" and "what would we do differently", is not applicable.
9 Project Retrospective

9.1 What went well

This has been a really interesting project, working with tools the team had never used before. Considering that neither of the team’s member had any experience working with the AWS tools or Ionic framework, improvements over the course of this project have been immense. Working as a two man team has had its advantages as all logistics have been rather easy. Communication with SmartGuide has been excellent. Team members have been in constant contact with Haukur Viðar Jónsson and Ægir Finnson, mostly through slack as well as few in person meetings, and have they been a great source of ideas and support.

9.2 What went badly

The work flow has been a lot less linear than initially planned. The inconsistency of work hours has not been ideal, as it resulted in some long sessions towards the end of the project. Work load from other classes interfered with the initial planned, as well as when the sprints were plan, the team didn’t take into account the exam period. Even though being a two man team had many advantages, fields were both team members are weak in, such as live presentations, might have been negated by having third or fourth team member, who could have balanced out those weaknesses. Also when one team member drops out for a period of time, half of the work hours for that period are lost. Few personal events came up that took a lot of time from team members.

9.3 What would we do differently

There would be few things done differently if the project was started today with the knowledge acquired from working on this project. Prioritizing what tasks should be completed first and a realistic estimation on how big some tasks actually are, was well off in some cases. With an open project like this, where implementation of nearly endless new features is possible, unrealistic expectations from team members of what can be achieved can get out of hand.
A Chatbot Inputs And Flow

**Intent** – An intent represents an action that the user wants to perform. Intents are triggered by utterances. An example of an utterance is: "Book a guide", which triggers the BookSmartGuide intent. Below intents are are indicated as follows: –**IntentName**, start with double dash and are in bold letters.

**Slot** - Slots are a part of an intents configuration. At runtime, Amazon Lex prompts the user for specific slot values. An example of this is if the user triggers the BookSmartGuide intent, lex will then ask the user to fill in the {Activity} slot with a valid input. Below slots are indicated with curly brackets around the slot name in question, {SlotName}.

–**GreetUser**:
User Input: “Who are you”
Response: “My name is Thor, and i’m your Smart guide assistant.”

–**HelloThere**
User Input: “Hi”, “Hello”
Response: “Hi, what can I help you with?”

–**WhatCanYouDo**
User Input: “What can you do”
Response: “I can help you find a guide for your dream trip. Just ask me, and I will help you.”

–**BookSmartGuide**
Response: “what activity are you looking for?”
User Input: {Activity}
Response: “in what country would you like to find a guide?”
User Input: {locationType}
Response: “what language should your guide speak?”
User Input: {LanguageType}
Response: “When would you like to book”
User Input: {Date}
Response: confirmation
User Input: “yes”, “no”  
Response:  
if yes: “Thank you, your booking request has been sent on. Our guides will get in contact with you as soon as possible”.  
if no: ”Your booking has not been placed”  

–BookSmartGuide alternative flow 1:  
User Input: “find a {Activity} guide that speaks {LanguageType}”  
Response: “in what country would you like to find a guide?”  
User Input: {locationType}  
Response: “When would you like to book”  
User input: {Date}  
Response: confirmation  
User Input: “yes”, “no”  
Response:  
if yes: “Thank you, your booking request has been sent on. Our guides will get in contact with you as soon as possible”  
if no: ”Your booking has not been placed”  

–BookSmartGuide alternative flow 2:  
User Input: “I would like to book a {Activity} guide in {locationType}”  
Response: “what language should your guide speak?”  
User Input: {LanguageType}  
Response: “When would you like to book”  
User input: {Date}  
Response: confirmation  
User Input: “yes”, “no”  
Response:  
if yes: “Thank you, your booking request has been sent on. Our guides will get in contact with you as soon as possible”  
if no: ”Your booking has not been placed”  

–BookSmartGuide alternative flow 3:  
User Input: “book a guide in {LocationType}”  
Response: “what activity are you looking for?”  
User Input: {Activity}  
Response: “what language should your guide speak?”  
User Input: {LanguageType}
Response: “When would you like to book”
User input: {Date}
Response: confirmation
User Input: “yes”, “no”
Response:
if yes: “Thank you, your booking request has been sent on. Our guides will get in contact with you as soon as possible”
if no: ”Your booking has not been placed”

–BookSmartGuide alternative flow 4:
User Input: “book a {LanguageType} speaking guide in {LocationType}”
Response: “what activity are you looking for?”
User Input: {Activity}
Response: “When would you like to book”
User input: {Date}
Response: confirmation
User Input: “yes”, “no”
Response:
if yes: “Thank you, your booking request has been sent on. Our guides will get in contact with you as soon as possible”
if no: ”Your booking has not been placed”