Bus Tracking App

by: Swole Team 6

SLACK CHANNEL: CSC-383.slack.com

Final Document

For: Professor Allison

On: 2/3/17

**Table of Contents**

**Abstract 3**

*1.1. Purpose of System 4*

*1.2. Scope of System 4*

*1.3. Development Methodology 4*

*1.4. Definitions, Acronyms, and Abbreviations 4*

*1.5. Overview of Document 5*

**2. Current System 6**

**3. Project Plan 6**

*3.1. Project Organization 6*

*3.2. Software and Hardware Requirements 7*

*3.3. Work Breakdown 7*

**4. Requirements of System 8**

*4.1. Functional and Nonfunctional Requirements 8*

*4.2. IDENTIFY Personas 8*

*4.3. Use Case Diagram 9*

*4.4. Requirements Analysis 10*

**5. Software Architecture 11**

*5.1. Overview 11*

*5.2. Subsystem Decomposition 11*

*5.3. Persistent Data Management 11*

**6. Object Design 12**

*6.1. Overview 12*

*6.2. Object Interaction 12*

*6.3. Detailed Class Design 12*

**7. Testing Process 13**

**8. Glossary 13**

**9. Appendix 13**

*9.1. Appendix A – Gantt Chart 13*

*9.2. Appendix B – Use Cases 13*

*9.3. Appendix C – User Interface Designs 13*

*9.4. Appendix D – Class Interfaces for Implemented Subsystems 13*

# 

# **Abstract**

Swole Team 6 proposes creating a bus tracking system that includes many features. Some of those features will be showing the route of a specific bus (selected by the user), the predicted arrival of the bus and if it is late, enable the students to log into and display this information in real time. The system will also maintain information about the parents and or students for contact information in case the need arises.

The application will also have the ability to track multiple busses at once. Any user can get the notifications of the buses whereabouts and if they are on schedule. The application will also allow the user to input whether a pickup is necessary or not and if the student is authorized to ride a specific bus.

**1. Introduction**

In this chapter, we will introduce the motivation for building the system

## ***1.1. Purpose of System***

The purpose of the system is to give students and parents a way to track their associated school bus and know what time it should arrive at each stop.

## ***1.2. Scope of System***

The system will allow parents to track the bus and get real time notification of when their children are being picked up from school and dropped off at home. The system will also allow the parent to view the bus driver’s information and vice versa so if problems arise, they can contact one another.

## ***1.3. Development Methodology***

Since agile is all about working with the business people to get exactly what they want throughout all phases of the project we have a few ways to do so. Currently one of our members has a family member that works as a bus driver and he is acting as our business person. One of our developers is working with him to figure out exactly what kinds of features he would want in a bus tracking software system.

Since then our developer has gotten a list of requirements from him along with some extras that our other developers came up with. At our next meeting, we are going to talk about all these requirements that are necessary to make this project tick along with some extras that are going to make it above and beyond expectation.

## ***1.4. Definitions, Acronyms, and Abbreviations***

Actors:External entities that interact with the system.

Agile:A method of development.

School:Educational institution for children.

ETA:Estimated Time of Arrival.

GPS:Global Positioning System.

## ***1.5. Overview of Document***

The grand scheme of this is to deliver an application that parents can install on their phones so they can pinpoint exactly when their children will be home.

# **2. Current System**

Not applicable

# **3. Project Plan**

## ***3.1. Project Organization***

**Phase 1 (add Dates)**

| Brian Freeman | Editor 1/16/2017-2/3/2017 |
| --- | --- |
| Dennis Kellogg | System Architect 1/16/2017-2/3/2017 |
| Luke Jeries | Leader 1/16/2017-2/3/2017 |
| Taylor Shephard, Lamis Alqashat | Minute Keeper 1/16/2017-2/3/2017 |
| Dan Wiseman | Secretary/Diary Keeper 1/16/2017-2/3/2017 |

**Phase 2 (add Dates)**

| Taylor Shephard | Leader 2/6/2017-2/17/2017 |
| --- | --- |
| Brian Freeman, Dan Wiseman | Validater/Architect 2/6/2017-2/17/2017 |
| Lamis Alqashat | Minute Keeper 2/6/2017-2/17/2017 |
| Luke Jeries | Diary Keeper/Tester 2/6/2017-2/17/2017 |
| Dennis Kellogg | Editor 2/6/2017-2/17/2017 |

## ***3.2. Software and Hardware Requirements***

Hardware: smartphone and the ability to download an application

Software:iOS, Android OS

## ***3.3. Work Breakdown***

| **Task #** | **Task** | **Description** | **Duration** | **Dependencies** |
| --- | --- | --- | --- | --- |
| 1 | Team Introduction | Get to know team members, consider strengths and weaknesses, brainstorm project topic | 7 days |  |
| 2 | Project Idea Development | Cultivate existing project idea, consider all potential uses, consider potential challenges | 7 days | 1 (M1) |
| 3 | Use Case Creation | Refine proposed features into clear use cases | 3 days |  |
| 4 | Use Case Completion | Submit Use Case diagram for review, consider future Contextual Use Case details | 4 days | 3 (D1) |
| 5 | Persona Research | Consider relevant users, begin consideration of Personas | 3 days |  |
| 6 | Documentation - SFD | SFD Stages 1-4.3 | 3 days | 2, 4 (M2) (D2) |
| 7 | Software Architecture | Divide Project into subsystems, identify objects, finalize member roles for development | 3 days | 6 |
| 8 | Object Design | Develop models into workable code/design | 14 days | 7 |
| 9 | Implementation Phase 1 | Begin coding and auxiliary software implementation | 20 days | 8 |
| 10 | Review of Implementation 1 | Members all reconvene for group reflection | 4 days | 9 (M3) |
| 11 | Implementation Phase 2 - Final | Finalize basic development | 14 days |  |
| 12 | "Hands on" Testing | Rigorous testing while working with key developer(known problems) | 7 days | 9, 11 (M4) (D3) |
| 13 | "Hands off" Testing | Testing with no insight from key developer(finding unknown problems) | 7 days |  |
| 14 | System Familiarity Development | Bring all members up to speed with status/details of sub systems | 2 days | 9, 11 |
| 15 | Creation of FD | Complete FD, begin Power Point | 20 days | (M5) (D4) |
| 16 | Final Presentation | Present and submit FD | 1 day | 15 |

M – Milestone D – Deliverable

# **4. Requirements of System**

The system we are proposing

## ***4.1. Functional and Nonfunctional Requirements***

Functional: App, GPS tracking, ETA Estimates, Contact Information, Map, All these features for all busses.

Non-functional: Show route on map, program to display bus info at school, Features to get ETA, Alarm Notification for student pickup and drop off, student to bus correlation, Ability to select other busses, RFID

## ***4.2. IDENTIFY Personas***

Parents: Busy people with tight schedules that need to know when their child or children will be leaving school and arriving home.

Drivers: Pick the kids up from school and drop them off at home. Need to be able to let parents know when the kids are being picked up and being dropped off. Along with alerting parents of emergencies

Students: The children being picked up and dropped off. They need to be able to check into the bus and Check out of the bus.

School Faculty: Work at the school and look out for the general wellbeing of the children. Need to know when the students get on the bus and when the students are dropped off the bus. Also need to be able to track the bus to see where its at and if has ran into any issues.

## ***4.3. Use Case Diagram***

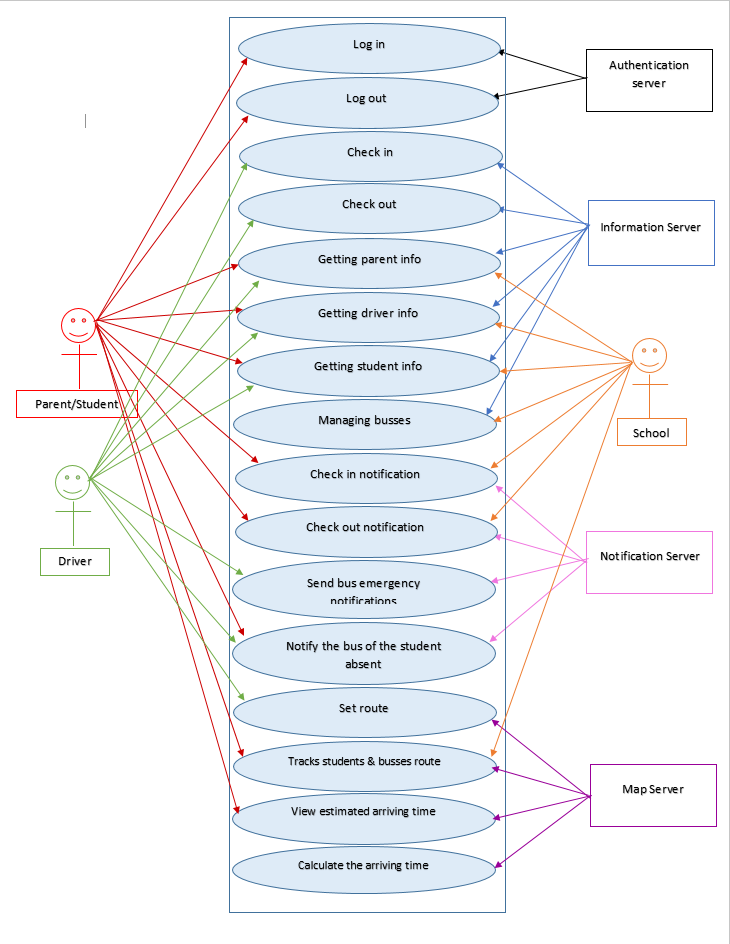
The next figure depicts the interaction between the actors and the previously described use cases. A description for each actor follows.

Parent: The main user of the application, guardian of the student.

Student: Rides the bus, could be in elementary/middle/high school

Driver: Operates the bus

School: Owner of the bus, manages the bus



*Figure 1: Use Case Diagram*

## ***4.4. Requirements Analysis***

# **5. Software Architecture**

.

## ***5.1. Overview***

## ***5.2. Subsystem Decomposition.***

## ***5.3. Persistent Data Management***

# **6. Object Design**

This chapter will build on the architecture and design.

## ***6.1. Overview***

## ***6.2. Object Interaction***

## ***6.3. Detailed Class Design***

# **7. Testing Process**

7.1. User Experience Tests

7.2. Systems Tests

7.3 Subsystems Tests

# **8. Glossary**

# **9. Appendix**

## ***9.1. Appendix A – Gantt Chart***

## ***9.2. Appendix B – Use Cases***

## ***9.3. Appendix C – User Interface Designs***

## ***9.5. Appendix D – Class Interfaces for Implemented Subsystems***