

**The University of the West Indies**

**Department of Computing and Information Technology**

**Undergraduate Project Course**

**Project Timeline, Work Breakdown and Use Case Guidelines**

**Project Name**

University Shuttle Routing and Tracking System

**Project Members**

Amanda Seenath 816002935

Azel Daniel 816002285

Michael Bristol 816003612

**Project Objectives**

The UWI SRTS project’s main aim is to produce an Android application that will enable students of the university to possess real time tracking information on the University’s shuttles from the palm of their hands. It will do this by tracking shuttle locations in real time and provide estimated times of arrivals to students based on that information.

**Work Breakdown Structure (Deliverables)**

The major deliverables for UWI SRTS project are:

1. Requirements Specification
2. Design Specification
3. Software Product
4. Testing and Documentation

These major deliverables can be further decomposed into the following smaller deliverables (work packages):

1. Requirements Specification
   1. Functional Requirements Specification
   2. Non-functional Requirements Specification
   3. Domain Requirements Specification
   4. User Requirements Specification
   5. System Requirements Specification
2. Design Specification
   1. Lo-fi prototypes
   2. Hi-fi prototypes
   3. Architecture Design Specification
   4. Context diagram
   5. Class diagram
   6. Use case diagram
   7. Sequence diagram
   8. Entity-relationship diagram for database
3. Implementation
   1. Implementation Document
   2. Iteration 1
   3. Iteration 2 and future works
4. Testing and Documentation
   1. Test plan

**Timeline Development (Schedule)**

The following are the tasks and activities that need to be completed to produce each work package:

1. Requirements Specification
   1. Functional Requirements Specification
      1. Requirements Elicitation (questionnaires and interviews)
      2. Analysis of questionnaire and interview responses
      3. Functional requirement specification based on response analysis
   2. Non-functional Requirements Specification
      1. Non-functional requirements specification based on response analysis
   3. Domain Requirements Specification
      1. Domain requirements specification based on response analysis
   4. User Requirements Specification
      1. User requirements specification based on response analysis
   5. System Requirements Specification
      1. System requirement specification bases on response analysis
2. Design Specification
   1. Lo-Fi Prototypes
      1. Brainstorm functional aspects of the system
      2. Produce sketches on the major functional aspects of the application
   2. Hi-Fi Prototypes
      1. Decide on Hi-Fi prototyping tool
      2. Choose the most relevant Lo-Fi prototypes to produce Hi-Fi prototypes from
      3. Produce Hi-Fi prototypes using the prototyping tool
   3. Architecture Design Specification
      1. Decide on appropriate architecture(s) based on the type of the application.
      2. Produce a diagram of the finalized application architecture
   4. Context Diagram
      1. Brainstorm entities to be included in the context diagram
      2. Produce context diagram consisting of those entities
   5. Class Diagram
      1. Brainstorm what concepts should be encapsulated as objects and classes
      2. Produce class diagram based on finalized classes
   6. Use Case Diagram
      1. Brainstorm common use cases of the application
      2. Describe these common use cases
      3. Produce use case diagram encapsulating these use cases
   7. Sequence Diagram
      1. Choose main concept to be encapsulated into a sequence diagram
      2. Produce sequence diagram from this main concept
   8. Entity Relationship Diagram
      1. Brainstorm firebase database schema
      2. Produce diagram of firebase database schema
3. Implementation
   1. Implementation Documentation
      1. Brainstorm Implementation Plan
      2. Assign estimated completion times to implementation pieces
      3. Draft implementation document
   2. Iteration 1
      1. Create base application: Implement a mock up Hi-Fi- In order to produce output from back end implementation.
      2. Setup firebase database: Create a team account for firebase operations
      3. Create listing of static routes: Using the information gathered from interviews, the route listings and locations would by inputted using Google Maps API
      4. Create detail of static route:
      5. Brainstorm model for static route detail in order to determine the most efficient way of that data storage
      6. Implement static route detail using information gathered from the existing system. This will include the integration of Google Maps API to the system’s information
      7. Authentication of students, drivers and administrators: Authentication of students, teachers and administrators- Using Email addresses and firebase, the students will be authenticated via a verification code. This process will be functionally tested and will be conducted internally by the team as well as by selected testers.
   3. Iteration 2 and future works
      1. Functionality for adding of routes dynamically by administrators
      2. Unit testing for user specified user operations using the mock- up H-FI for output
      3. Test dynamic changes made by administrators
      4. Implement administrator changing routes
      5. Implement administrators changing routes dynamically
      6. Functionality for adding and managing driver accounts:
      7. Implement adding a driver account
      8. Implementing functionality of roles
      9. Associate role with operations for an administrator
      10. Notifications for students when bus has arrived
      11. Implementation of miscellaneous improvements
4. Testing and Documentation
   1. Test Plan
      1. Brainstorm testing scenarios that may occur
      2. Conduct alpha testing with small select group of testers
      3. Conduct beta testing with larger pool of testers.

**Task Resources**

The following is a list of identified resources for each task:

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Time** | **Knowledge** | **Monetary Costs** |
| Requirements Elicitation (questionnaires and interviews) | 10 | Minimal | Low |
| Analysis of questionnaire and interview responses | 4 | Minimal | None |
| Functional requirement specification based on response analysis | 1 | Moderate | None |
| Non-functional requirements specification based on response analysis | 1 | Moderate | None |
| Domain requirements specification based on response analysis | 1 | Moderate | None |
| User requirements specification based on response analysis | 1 | Moderate | None |
| System requirement specification bases on response analysis | 1 | Moderate | None |
| Brainstorm functional aspects of the system | 1 | Low | None |
| Produce sketches on the major functional aspects of the application | 8 | Low | None |
| Decide on Hi-Fi prototyping tool | 1 | Low | None |
| Choose the most relevant Lo-Fi prototypes to produce Hi-Fi prototypes from | 1 | Moderate | None |
| Produce Hi-Fi prototypes using the prototyping tool | 2 | Moderate | None |
| Decide on appropriate architecture(s) based on the type of the application. | 2 | Low | None |
| Produce a diagram of the finalized application architecture | 3 | Moderate | None |
| Brainstorm entities to be included in the context diagram | 2 | Moderate | None |
| Produce context diagram consisting of those entities | 3 | Moderate | None |
| Brainstorm common use cases of the application | 1 | Moderate | None |
| Describe these common use cases | 2 | Moderate | None |
| Produce use case diagram encapsulating these use cases | 2 | Moderate | None |
| Choose main concept to be encapsulated into a sequence diagram | 2 | Moderate | None |
| Produce sequence diagram from this main concept | 3 | Moderate | None |
| Brainstorm firebase database schema | 2 |  | None |
| Produce diagram of firebase database schema | 3 | Moderate | None |
| Brainstorm Implementation Plan | 5 | Moderate | None |
| Assign estimated completion times to implementation pieces | 1 | Moderate | None |
| Draft implementation document | 5 | Moderate | None |
| Create base application: Implement a mock up Hi-Fi- In order to produce output from back end implementation. | 2 | Moderate | None |
| Setup firebase database: Create a team account for firebase operations | 2 | A lot | None |
| Create listing of static routes: Using the information gathered from interviews, the route listings and locations would by inputted using Google Maps API | 2 | Moderate | None |
| Create detail of static route | 2 | Moderate | None |
| Brainstorm model for static route detail in order to determine the most efficient way of that data storage | 1 | Moderate | None |
| Implement static route detail using information gathered from the existing system. This will include the integration of Google Maps API to the system’s information | 3 | Moderate | None |
| Authentication of students, drivers and administrators: Authentication of students, teachers and administrators- Using Email addresses and firebase, the students will be authenticated via a verification code. This process will be functionally tested and will be conducted internally by the team as well as by selected testers. | 11 | A lot | None |
| Functionality for adding of routes dynamically by administrators | 6 | A lot | None |
| Unit testing for user specified user operations using the mock- up H-FI for output | 2 | Low | None |
| Test dynamic changes made by administrators | 2 | Low | None |
| Implement administrator changing routes | 5 | Moderate | None |
| Implement administrators changing routes dynamically | 5 | Moderate | None |
| Functionality for adding and managing driver accounts | 5 | A lot | None |
| Implement adding a driver account | 5 | A lot | None |
| Implementing functionality of roles | 5 | A lot | None |
| Associate role with operations for an administrator | 5 | Moderate | None |
| Notifications for students when bus has arrived | 5 | A lot | None |
| Implementation of miscellaneous improvements | 3 | Moderate | None |
| Brainstorm testing scenarios that may occur | 2 | Moderate | None |
| Conduct alpha testing with small select group of testers | 4 | Low | Low |
| Conduct beta testing with larger pool of testers. | 5 | Low | Low |

**Gantt Chart**

The following is a Gantt chart depicting the sequence of the various activities and tasks. It also models a critical path showing what tasks are dependent on others.

(See next page)