

The University of the West Indies

Department of Computing and Information Technology

Undergraduate Project Course

**Project Implementation Document**

Project Name

UWI Shuttle Routing and Tracking System

Project Members

Amanda Seenath 816002935

Azel Daniel 816002285

Michael Bristol 816003612

Architectural Design

A close up of text on a white background

Description automatically generated

Figure 1. Component Diagram for the UWI SRTS application. Click [here](https://myuwi-my.sharepoint.com/:i:/g/personal/amanda_seenath_my_uwi_edu/Ecav_fgZPO9ErdToGRu0B4MBkiN92AqXQQmaCbXXm0hl2g?e=wpuEUo) to open a link to view Figure 1.

The description of the components in the system is as follows:

|  |  |  |
| --- | --- | --- |
| Component | Description | Purpose |
| Route Screen | UI screen that allows users to view and, with appropriate authorisation, request manipulation of a single route object. | Its main purpose is to provide an interface that allows access of a single route object to a user. |
| Routes Screen | UI screen that allows users to view a list of route objects. | Its main purpose is to provide an interface that allows access of a list of route objects to a user. |
| User Screen | UI screen that allows users to view and, with appropriate authorisation, request manipulation of a single user object. It also allows a user to login and signup. | Its main purpose is to provide an interface that allows access of a single user object to an authorised user. |
| Users Screen | UI screen that allows users, with proper authorisation, to view a list of user objects. | Its main purpose is to provide an interface that allows access of a list of user objects to an authorised user. |
| Vehicle Screen | UI screen that allows users to view and, with appropriate authorisation, request manipulation of a single vehicle object. | Its main purpose is to provide an interface that allows access of a single vehicle object to a user. |
| Vehicles Screen | UI screen that allows users to view a list of vehicle objects. | Its main purpose is to provide an interface that allows access of a list of vehicle objects to a user. |
| Alert Screen | UI screen that allows users to view and, with appropriate authorisation, request manipulation of a single alert object. | Its main purpose is to provide an interface that allows access of a single alert object to a user. |
| Alerts Screen | UI screen that allows users to view a list of alert objects. | Its main purpose is to provide an interface that allows access of a list of alert objects to a user. |
| Route Controller | This component takes commands from the user, sends update commands to the route model and sends update instructions to the route view (screen). | The main purpose of this component is to decouple the route view (screen) from the route model. This allows for a more flexible application. |
| Routes Controller | This component takes commands from the user and sends update instructions to the routes view (screen). | The main purpose of this component is to decouple the routes view (screen) from the routes model. |
| User Controller | This component takes commands from the user, sends update commands to the route model and sends update instructions to the user view (screen). | The main purpose of this component is to decouple the user view (screen) from the user model. |
| Users Controller | This component takes commands from the user and sends update instructions to the users view (screen). | The main purpose of this component is to decouple the users view (screen) from the users model. |
| Vehicle Controller | This component takes commands from the user, sends update commands to the route model and sends update instructions to the vehicle view (screen). | The main purpose of this component is to decouple the vehicle view (screen) from the vehicle model. |
| Vehicles Controller | This component takes commands from the user and sends update instructions to the vehicles view (screen). | The main purpose of this component is to decouple the vehicles view (screen) from the vehicles model. |
| Alert Controller | This component takes commands from the user, sends update commands to the route model and sends update instructions to the alert view (screen). | The main purpose of this component is to decouple the alert view (screen) from the alert model. |
| Alerts Controller | This component takes commands from the user and sends update instructions to the alerts view (screen). | The main purpose of this component is to decouple the alerts view (screen) from the alerts model. |
| Route | This component models information for a single route in the system. | The main purpose of this component is to model information for a single route and sync that information to the database. |
| Routes | This component models information for a list of routes in the system. | The main purpose of this component is to model information for a list of routes and sync that information to the database. |
| User | This component models information for a single user in the system. | The main purpose of this component is to model information for a single route and sync that information to the database. |
| Users | This component models information for a list of users in the system. | The main purpose of this component is to model information for a list of users and sync that information to the database. |
| Vehicle | This component models information for a single vehicle in the system. | The main purpose of this component is to model information for a single route and sync that information to the database. |
| Vehicles | This component models information for a list of vehicles in the system. | The main purpose of this component is to model information for a list of vehicles and sync that information to the database. |
| Alert | This component models information for a single alert in the system. | The main purpose of this component is to model information for a single route and sync that information to the database. |
| Alerts | This component models information for a list of alerts in the system. | The main purpose of this component is to model information for a list of alerts and sync that information to the database. |
| Database | This component is the secondary storage for the models. | The main purpose of this component is to save model information. |

Class Diagram

A screenshot of a cell phone

Description automatically generated

Figure 2. Class Diagram for the UWI SRTS Application. Click [here](https://myuwi-my.sharepoint.com/:i:/g/personal/amanda_seenath_my_uwi_edu/ERhZ_jC9T_RKlb54vhxUXK4BbTY9BvOV2hL9yvXxS_eAhA?e=aDmsZ7) to open a link to view Figure 2.

Entity Relationship Model

(See next page)

A screenshot of a cell phone

Description automatically generated

Figure 3. Entity relationship Diagram for the UWI SRTS Firebase database.

User Scenarios and Sequence Diagrams

1. Goal: To successfully sign up

Primary Actor: The Primary User

Conditions: The user enters valid account information

Outcome: The user is alerted that the sign up was successful.

Starting situation: The user is greeted with a welcome screen

Concurrent activities: None

End situation: User now has a user account

Things that can go wrong: User entering incorrect data.

Normal flow of events:

1. The primary user enters a valid email, password, name and student Id.
2. The Screen collects this information and forwards it to Authentication service
3. The Authentication service verifies the inputted user information, and this is not visible to the user. the information to its local device Without storage, the schedule cannot be saved.
4. After verification from the authentication service, the account information is sent to the firebase database to be stored.
5. Store the account information. In the firebase database.

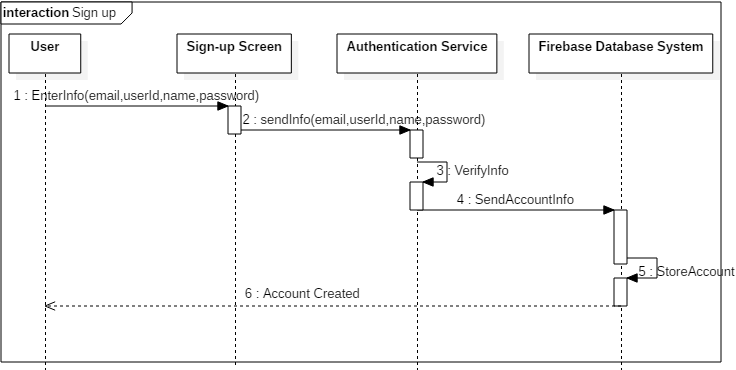


Figure 4: Sequence diagram of a user successfully singing up

1. Goal: To successfully track the real-time location of shuttle

Primary Actor: The Primary User

Conditions: The shuttle is moving and on-duty, the user is logged in  
Outcome: The user views the location of the shuttle   
Concurrent activities: None  
End situation: The shuttle is viewed

Things that can go wrong: Time Service

Normal flow of events:

1. The primary User selects as shuttle from the initial list presented on the screen
2. Each Shuttle has a unique Id. After the user selects a shuttle. The Shuttle Id is sent to firebase SPS Firebase
3. Using the Unique Identifier of Shuttle Id a shuttle is located in the firebase database
4. The Service will be displayed in real time and showing the exact location of a shuttle hence we would have a time in which we request updates from automatically.
5. The updated location is posted on GPS firebase
6. Display the new shuttle location on google map for the user

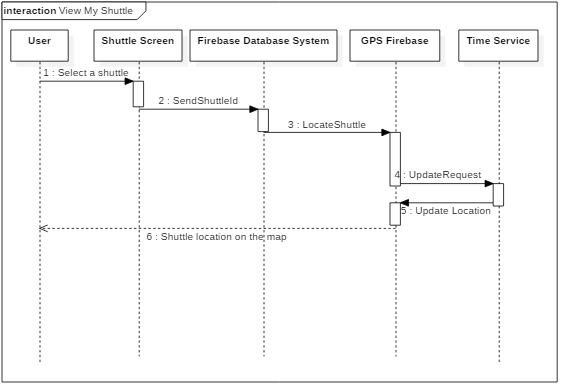


Figure 5: Scenario of user successfully tracking the real-time location of a shuttle

Description of Technology Used

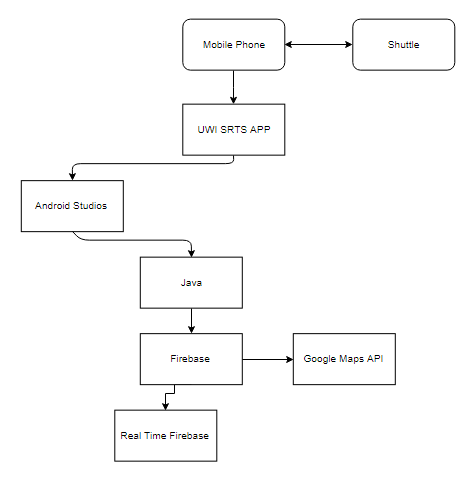


Figure 6. The integration of technologies for the UWI SRTS application.

|  |  |  |
| --- | --- | --- |
| Technology Type | Hardware/ Software | Descriptions |
| Shuttle | Hardware | This is vehicle which transports the students. |
| Mobile Phones | Hardware | This is a device issued to each driver. The drive will have the features of location finding and will contain a version of the UWI SRTS. |
| Java | Software Programming Language | This is a programming language that is used by Android Studios and is used to implement all software for the application. |
| Android Studio | Software | This is an Integrated development environment that is used specifically for Google Android operating system. |
| UWI SRTS | Software | This is the application that can be downloaded to Android devices. |
| Firebase | Software | Application Development Platform which offers database services to this application. |
| Firebase Real Time | Software | This is a branch of Firebase’s database services which allows the current shuttle locations to be displayed across multiple devices synchronously. |
| Google Maps API | Software | This is an Application Programming Interface (API) which is used to monitor static locations on Google and to track the location of shuttles. |