



# STACKS ON STACKS

Pushing The Best Solutions

Akua Afrane-Okese 15019773

Ntiko Mathaba 14012503

Tyler Matthews 15302424

With contributions from Linda Zwane (1419946)

# Functional and requirements document

Stacks On Stacks

## Contents

1.	System overview .....	2
1.1	Purpose .....	2
1.2	Background .....	2
1.3	Project Scope .....	2
1.4	Definitions, acronyms and abbreviation .....	2
1.5	UML Domain Model .....	3
2.	Functional Requirements .....	4
2.1	Users .....	4
2.2	Subsystems .....	5
2.2.1	User Management Module .....	6
2.3	Specific Requirements .....	8
2.3.1	Requirements traceability matrix .....	9
3	Non-functional requirements .....	12
4	System architecture .....	12
4.1	Interfaces .....	12
4.1.1	User interfaces .....	12
4.1.2	Hardware interfaces .....	13
4.2	Architectural Styles .....	13

## 1. System overview

### 1.1 Purpose

The purpose of this document is to give a business overview of ArivlApp. The proposed system is an app that will improve estate management systems for residential areas and office blocks. The app is going to permit estate residents authorized, seamless access using only their smart phones. The web application is going to allow estate managers to manage all the users of the estate electronically.

### 1.2 Background

Currently, boom gate access control system requires a car user to either use a token to gain access or by the use of a remote. This is a problem because in a case where there are a large number of cars that either want to exit or enter a premises, there exists a delay which results in traffic. This is a problem because the delay in time can be avoided.

### 1.3 Project Scope

ArivlApp is aimed at giving access to authorized users at registered boom gates. A user needs to be registered in order to gain access to certain boom gates. The user logs in using their phone number and password only once. The user needs to enable their Bluetooth to ensure that the boom gate opens when in close proximity.

Other functionality may include, but not limited to, turning on the lights and the air-conditioner inside the house or office block as soon as the user enters the estate after having opened the boom gate using the ArivlApp.

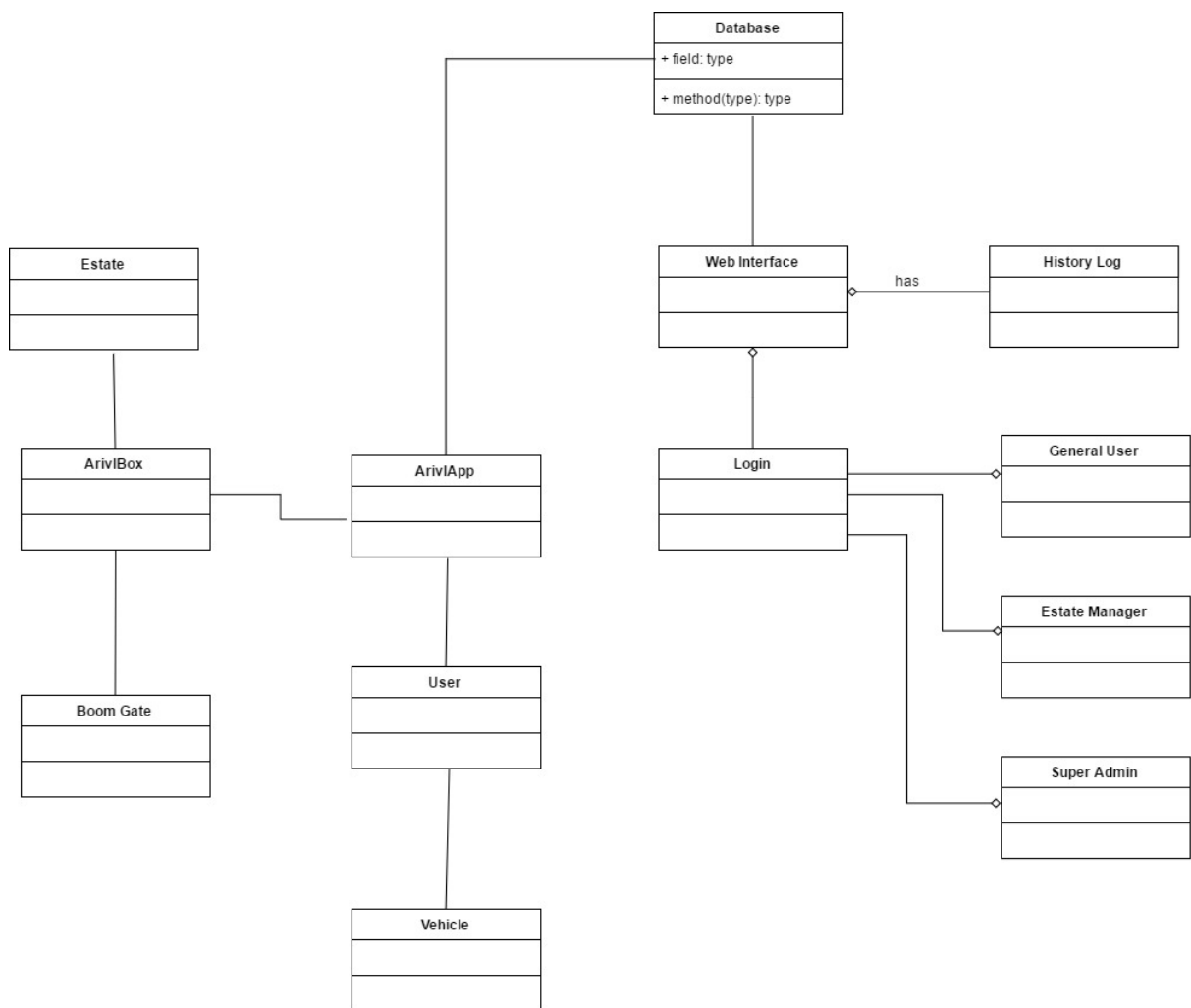
### 1.4 Definitions, acronyms and abbreviation

BLE: Bluetooth Low Energy

App: Mobile Software Application

Admin: Administrator

## 1.5 UML Domain Model



## 2. Functional Requirements

### 2.1 Users

User	Description	Privileges
<b>Super Admin</b>	The Super Admin user has the highest level of access and responsibility above a regular estate administrator, and has complete control of the entire system.	<ul style="list-style-type: none"><li>• Approve estate applications</li><li>• Manage estates</li><li>• Manage estate managers and system users</li></ul>
<b>Estate Admin</b>	The estate admin/manager has privileges which include modifying access privileges for other system users, inviting and removing users and changing member roles. A system can have more than one admin.	<ul style="list-style-type: none"><li>• Approve system user applications</li><li>• Manage system users</li></ul>
<b>System User</b>	A user is a member who can use the system for which access privileges have been granted and can invite other users such as family members and guests. A system can have multiple users, invited by an admin or the owner.	<ul style="list-style-type: none"><li>• Manage guest users</li><li>• Add other users (Family Members)</li><li>• Manage profile</li></ul>
<b>Guest</b>	A default set of permissions and privileges given to non-registered users of a system or service. No information is stored for the guest user. When accessing the service, the user assumes the guest role without being logged in. The guest user may use public services and my register or log in.	<ul style="list-style-type: none"><li>• 24 Hour Access to the estate</li></ul>

## 2.2 Subsystems

The core of the system is an access control tool which is enriched with functionality that allows system users to add other users. The high level modules and their responsibilities are shown in figure 1 down below.

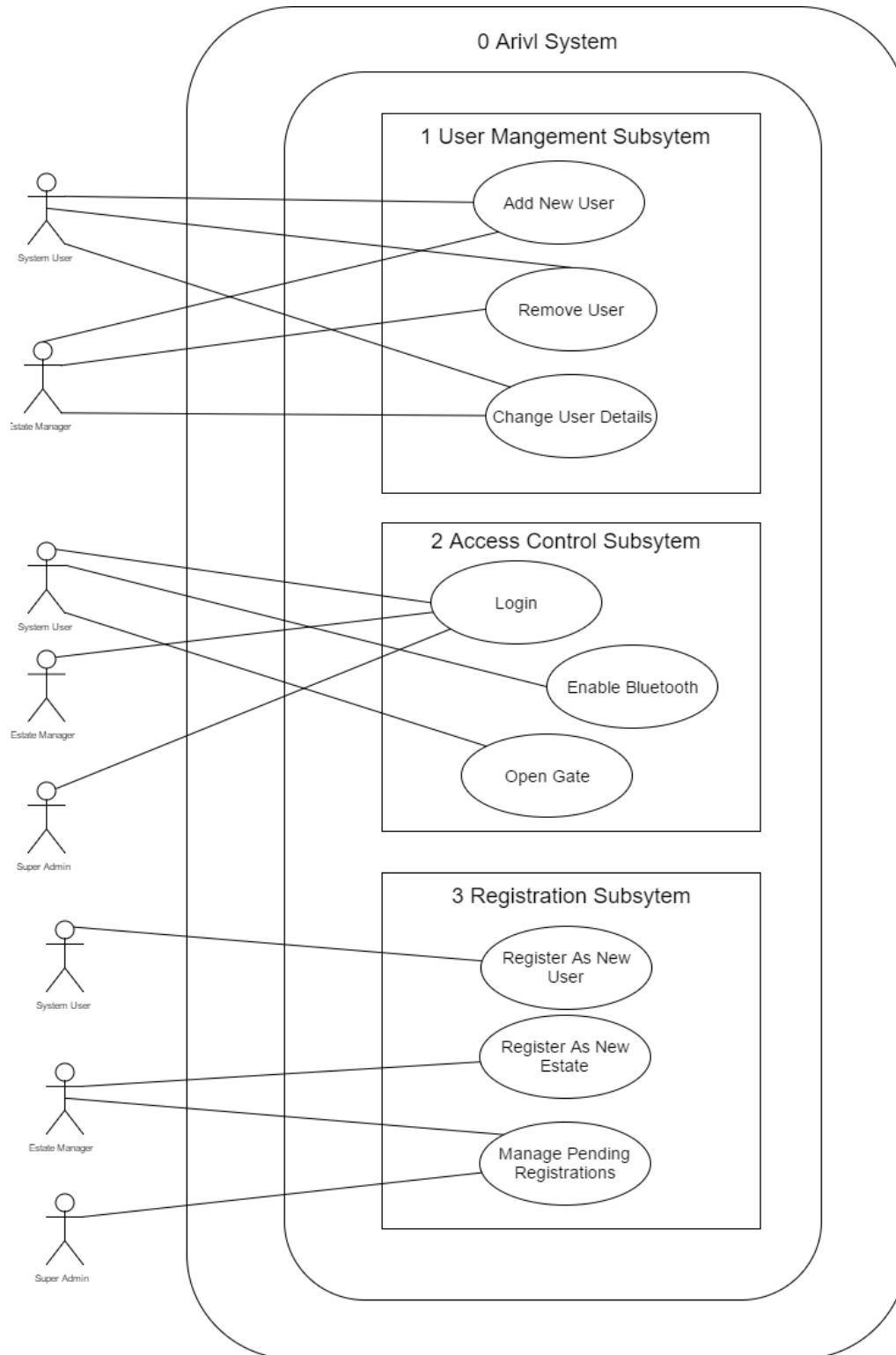


Figure 1 Subsystems Diagram

### 2.2.1 User Management Module

The scope of the user management module is show in the figure down below. The user management module is responsible for maintaining information about the registered users of the system, including the authorities of each user. Administrators can manage information about the estate whilst users who have signed up may request services from the various modules and persist private information related to particular services.

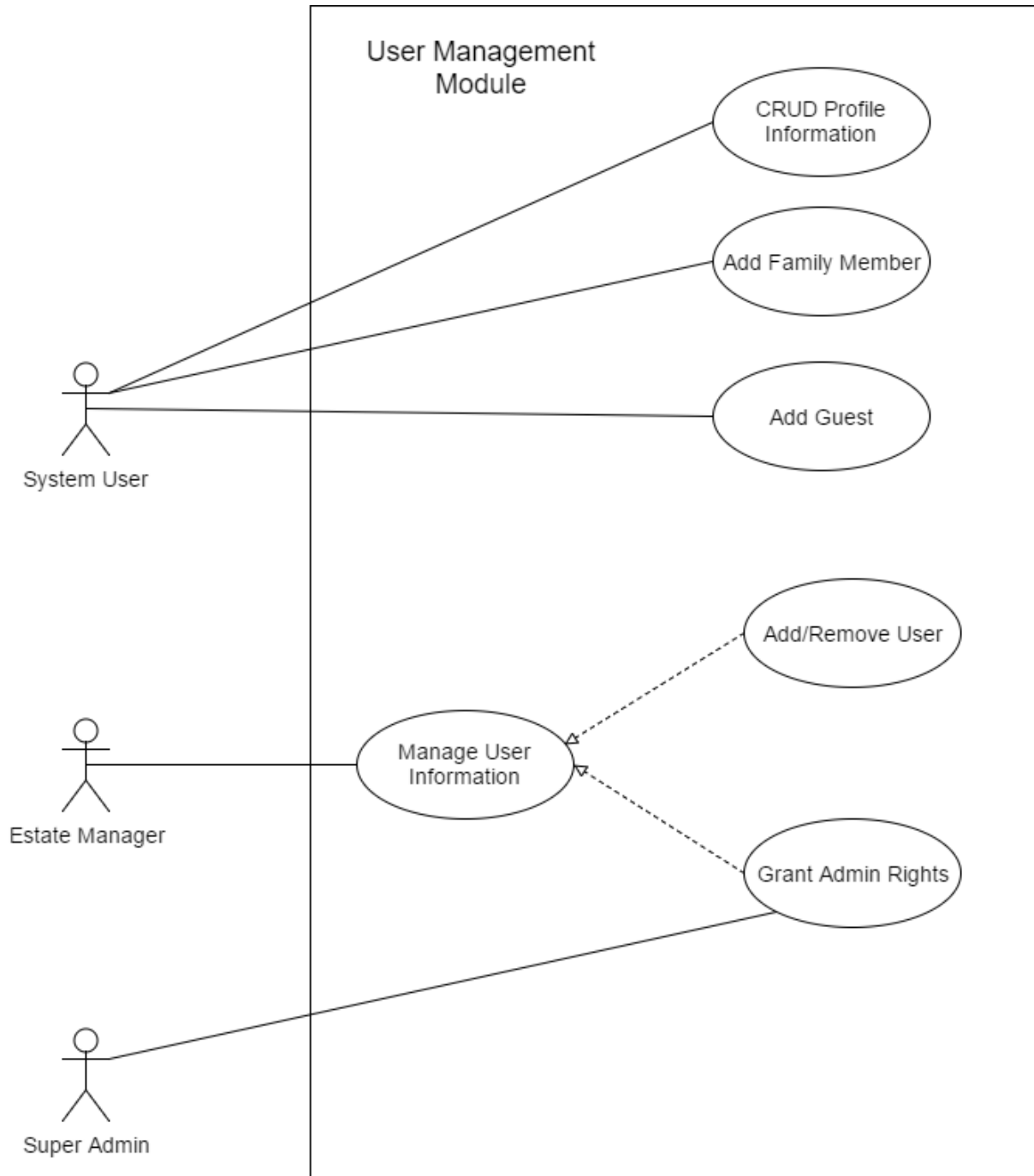


Figure 2: User Management Module

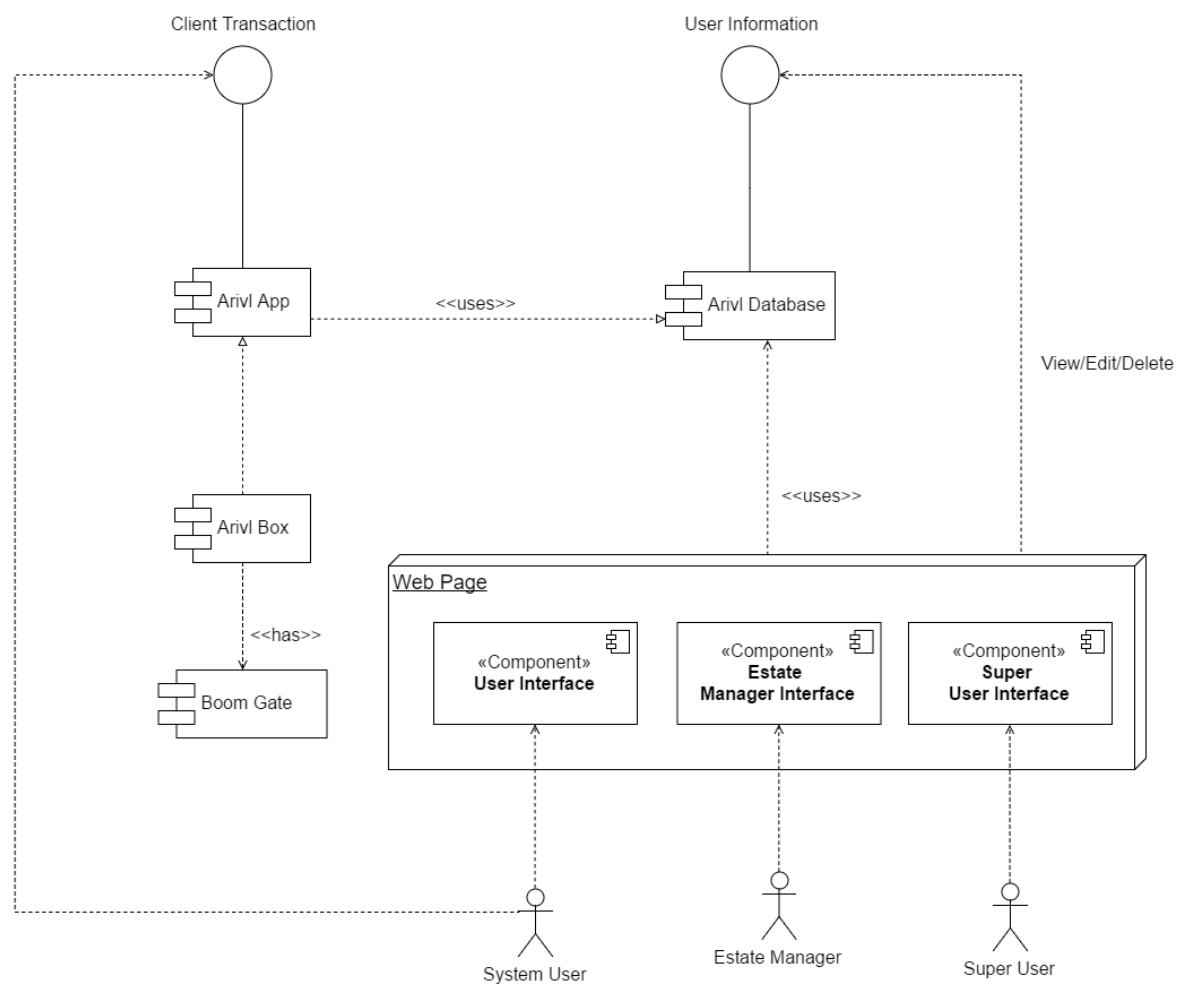


Figure 3: Arivl Component Diagram



## 2.3 Specific Requirements

Requirement ID	Requirement Statement
FR01	<ul style="list-style-type: none"><li>1.1 The user shall register<ul style="list-style-type: none"><li>1.1.1 The system user shall register with an estate</li><li>1.1.2 The estate shall register with Arivl (Super Admin)</li></ul></li><li>1.2 The user shall manage pending registrations<ul style="list-style-type: none"><li>1.2.1 The estate manager shall approve pending registrations</li><li>1.2.2 The estate manager shall decline pending registrations</li></ul></li></ul>
FR02	<ul style="list-style-type: none"><li>2.1 The user shall have a smart phone<ul style="list-style-type: none"><li>2.1.1 The user shall have a smart phone with an Android Operating System</li><li>2.1.2 The user shall have a smart phone with Bluetooth Technology</li></ul></li></ul>
FR03	<ul style="list-style-type: none"><li>3.1 The user shall have access granted and open the gate<ul style="list-style-type: none"><li>3.1.1 The user shall log on to the app to gain access to the estate</li><li>3.2.2 The user shall have Bluetooth enabled to gain access to the estate</li><li>3.2.3 The user shall be in close proximity to gain access to the estate</li></ul></li><li>3.2 The guest user shall have access for 24 hours only</li></ul>
FR04	<ul style="list-style-type: none"><li>4.1 The user shall manage his profile<ul style="list-style-type: none"><li>4.1.1 The user shall add family members<ul style="list-style-type: none"><li>4.1.1.1 The user shall add a vehicle</li></ul></li><li>4.1.2 The user shall add guests</li><li>4.1.3 The user shall remove family members</li><li>4.1.4 The user shall update his details</li></ul></li><li>4.2 The estate manager shall manage the estates details</li></ul>
FR05	<ul style="list-style-type: none"><li>5.1 The user shall log history<ul style="list-style-type: none"><li>5.1.1 The user shall log his history</li><li>5.1.2 The estate manager can log users history</li><li>5.1.3 The super admin shall log users history</li></ul></li></ul>
FR06	<ul style="list-style-type: none"><li>6.1 The user may retrieve forgotten password</li></ul>

### 2.3.1 Requirements traceability matrix

Requirement ID	User Management Subsystem	Access Control Subsystem	Registration Subsystem
<b>FR01</b>			
1.1			X
1.1.1			X
1.1.2			X
1.2	X		
1.2.1	X		
1.2.2	X		
<b>FR03</b>			
3.1			
3.1.1		X	
3.1.2		X	
3.1.3		X	
3.2		X	
<b>FR04</b>			
4.1	X		
4.1.1	X		
4.1.1.1	X		
4.1.2	X		
4.1.3	X		
4.1.4	X		
4.2	X		
<b>FR05</b>			
5.1	X		
5.1.1	X		
5.1.2	X		
5.1.3	X		
<b>FR06</b>			
6.1		X	

## UC: Tenant registers at an estate

Precondition: The user should own a house at the estate	
Actor: Tenant	System: Estate Admin
	0. System displays welcome page
1. Tenant clicks on the sign up link on the welcome page	2. System displays a web page with the form
3. Tenant fills in his details and submits the form	4. System prompts the user to enter the address of the estate
5. The user enters the estate address	6. System displays "Your application has been sent and you will be contacted once it has been approved"
7. Tenant sees the "Your application has been sent" message	
Post condition: NONE	

## UC: User adds a guest user

Precondition: The user should be registered to add a guest	
Actor: Tenant	System: Web App
	0. Web App displays home page
1. Tenant clicks on the log in link on the welcome page	2. Web App displays the log in page
3. Tenant fills in phone number and password	4. System sends the user the OTP
5. The user enters the OTP	6. System displays the form to capture guest details
7. Tenant fills in the guests information	8. System displays "Your guest has been added"
9. Tenant sees the "Your guest has been granted 24 hours access" message	
Post condition: NONE	

## UC: User opens gate

Precondition: The user should be in close proximity	
Actor: Arivl App	System: Arivl Box
	0. Arivl Box emits Bluetooth signal
1. App connects to the Arivl Box using Bluetooth Low Energy	2. Arivl App verifies the users details
	3. a) Arivl Box grants access and opens gate b) Arivl Box denies access to the user
4. Arivl App notifies the user of the outcome	5. Arivl Box ends communication with Arivl App
Post condition: NONE	

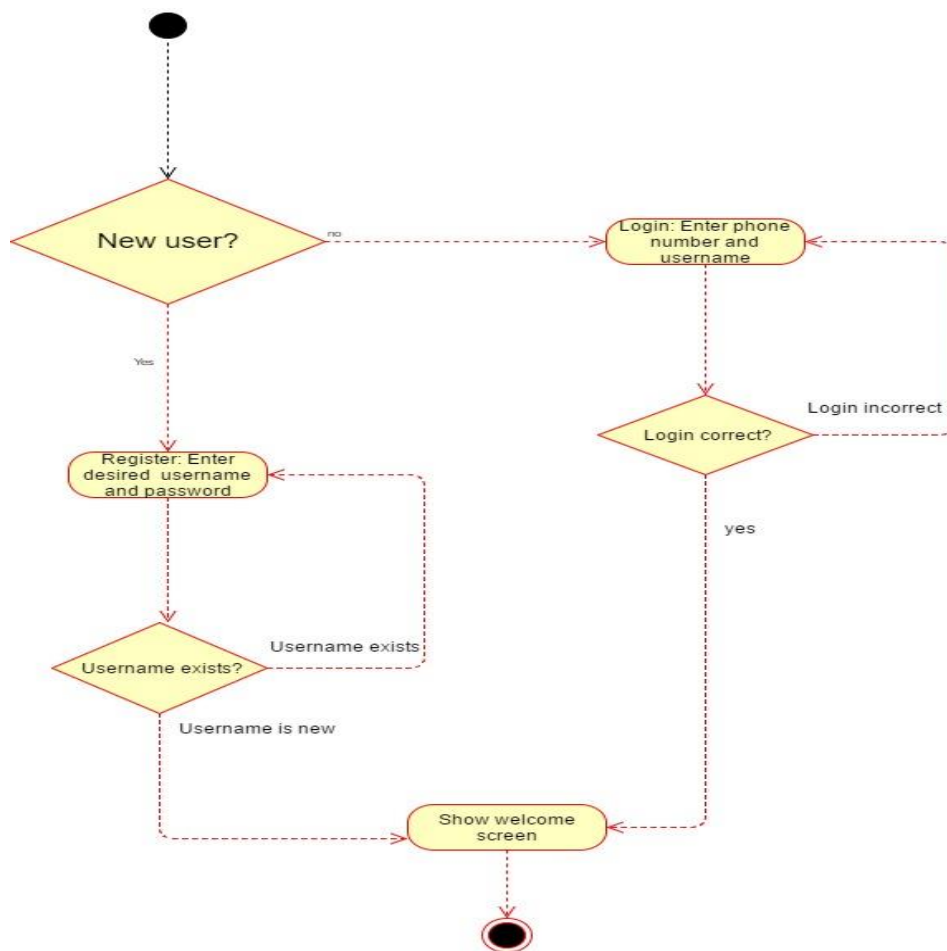


Figure 4: Login Activity

**User Login Activity:** When a user is not registered as part of the system, the fields shown in the domain model are stored for the user, using a unique automatically assigned ID. The user provides all other fields. The password should be stored in encrypted format. The user may change the value of any of the fields of his/her own record except the value of the isAdmin field. The difference between a Main user and an Admin is the value of the isAdmin field. Only Admin users may change the value of the isAdmin field of other users.

### 3 Non-functional requirements

Quality Requirements of the Arivl System: should include but not limited to the following

Requirement ID	Requirement Statement	Comments
NFR01	Performance	
NFR02	Security	
NFR03	Scalability	
NFR04	Usability	
NFR05	Availability	

### 4 System architecture

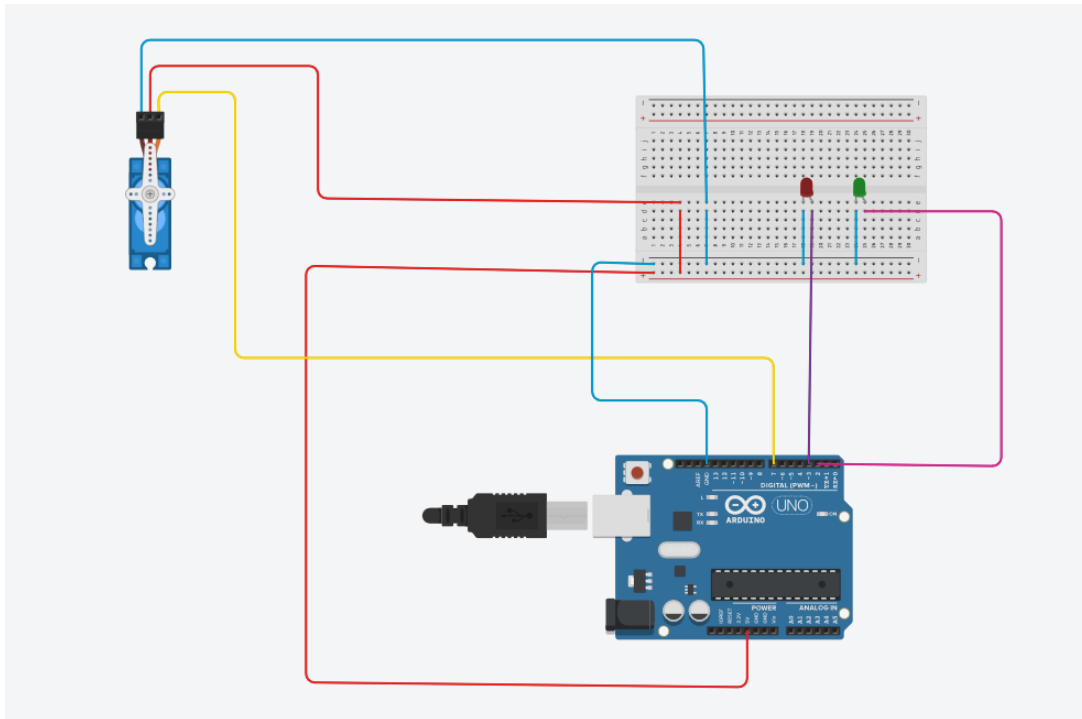
#### 4.1 Interfaces

##### 4.1.1 User interfaces

As specified in the high level scope diagram above, there are three concerns for the accessibility of the system to both normal users and administrator-level users: Android and Web-front end. Android refers to mobile apps and the Web-Front end refers to the ability of a user to use the system from a web browser from a device such as a tablet or smartphone. These are the two access channels through which users must be able to interface with the system.

User Interface	Description
System User	The system user can access the system through the Android application and the web front end.
Estate Manager	The estate manager has privileges which include modifying access privileges for other system users, inviting and removing system users. The estate manager can access the above mentioned functionality through the use of the web front end.
Super User	The Super Admin user has the highest level of access and responsibility above a regular estate administrator, and has complete control of the entire system. The super user can access the system through the web front end.

### 4.1.2 Hardware interfaces



Our system uses an Arduino Circuit board to simulate the boom gate belonging to the estate. The Arduino board has C++ code embedded on it that allows for communication between the Arivl app and the gate.

The Arduino board is connected to a BLE module connected on it which emits signals for users to receive via the Arivl app. When in close proximity, the Arivl app will attempt to connect to the board and verify whether the user has access or not.

## 4.2 Architectural Styles

At the highest level granularity, the ArivlApp system is based on the N-Tier Architecture. The second level of granularity can be visualized as to be based on the Event-Driven Architecture. The N-Tier architecture puts emphasis on having layers which are assigned to do a specific task(s). The apparent layers are: presentation layer, business layer and the persistence layer.

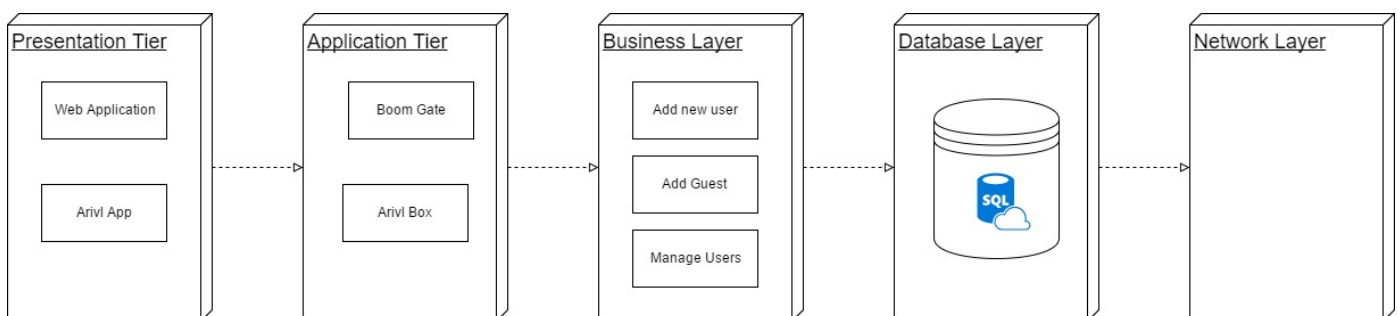


Figure 5: Arivl N-Tier Architecture

The presentation layer is responsible for the user interface of the application on the Arivl App and the website. The different users that have been identified in this document each have an interface that has been built for them. The user can send requests using the applications presentation layer.

The

At the second level of granularity, our system follows an event-driven architecture where the system depicts state dependent behaviour. The ArivlApp, firstly tries to connect to the ArivlBox using Bluetooth Low-Energy(BLE); if the user has access to that specific boom gate and if the ArivlBox is not connected to any other device, a button appears on the home screen of the application reading "Open Gate", which, when pressed, verifies the user's access to that specific boom gate and if the user is authorized access, the application sends a signal to the gate, requesting it to open the gate; consequentially, disconnects the user's ArivlApp from the ArivlBox. Furthermore, changing its state from "closed" to "open" and hence, opening the gate. Moreover, as soon as the gate is opened, a timer for 5 seconds is started, as the time stops, the state of the gate is changed to "closed" and the gate is closed. This allows another authorized device to connect to the ArivlBox as soon as possible.

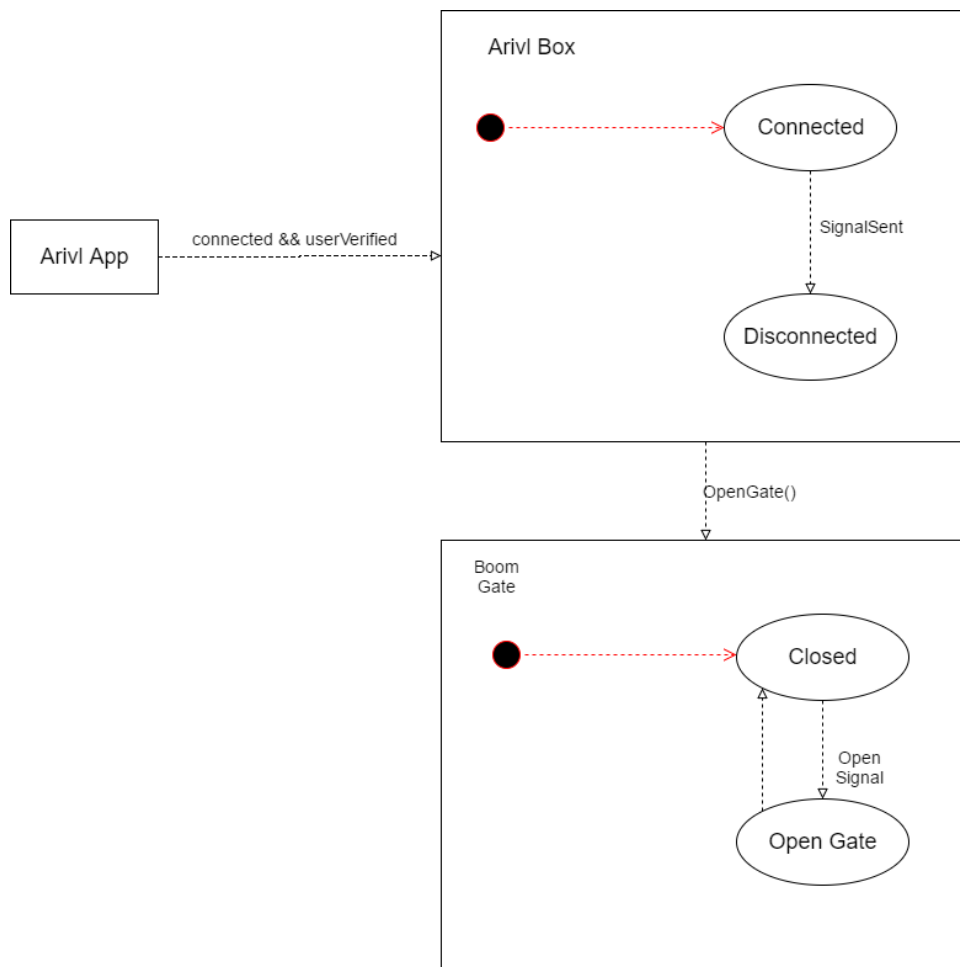


Figure 6: Event Driven Architecture

