INTRODUCTION

Our project was to develop a Smart Bicycle Lock for the Bicycle Sharing System currently in the campus called “Greenolution”. Smart Bicycle Lock as the name suggests is a lock which provides various other functionalities like ride tracking, ride halting facility, cashless ride payment, 24x7 functioning of the system, anti-theft alarm, issuing a ride using RFID card etc.

# KEY COMPONENTS/SPECIFICATIONS OF SMART LOCK

This diagram shows the basic schematic diagram of the circuit.

We used LinkIt ONE board in our smart lock which is GPS/GPRS/Bluetooth/Wi-Fi enabled. It also has some RAM and FLASH memory of its own. Detailed specifications of the LinkIt ONE board can be seen at <https://www.seeedstudio.com/LinkIt-ONE-p-2017.html>.

Apart from this we used separate RFID module MFRC-5222 for reading the RFID card.

This diagram shows the connection between LinkIt ONE board and the RFID module.

Other components includes:

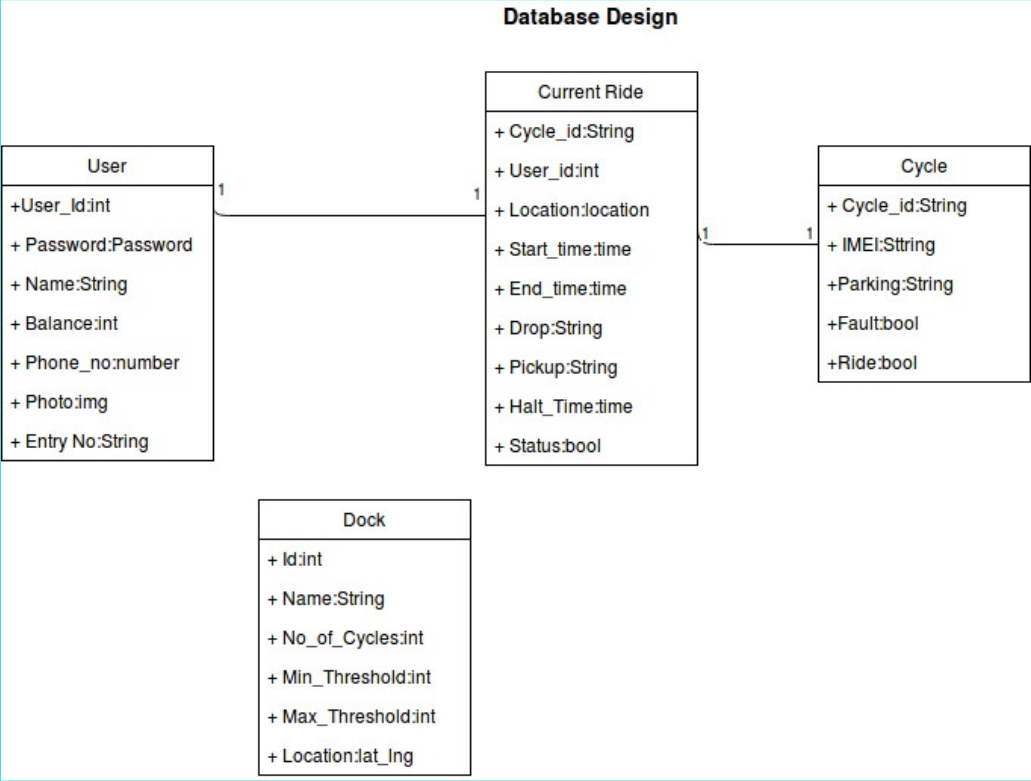
* Buzzer: For anti-theft alert
* LED: To display the busy state of lock (when the user is not allowed to scan the RFID card)
* Motor: For unlocking the locking mechanism.
* Battery: A 9V battery to give power to the motor.
* Limit Switch: To get the lock/unlock state of the mechanism.
* Push Button: To differentiate between ending and halting of a ride.

Entities involved in various steps of cycle sharing process are

* Device: The main lock containing all the hardware part for eg. components discussed earlier, circuit etc.
* Server: Sends commands to and receives commands from the device and executes them by accessing/altering the data in database.
* Database: Contains all the data required in a cycle sharing process. The data is organised in different tables like user, lock, ride and dock. This data can be accessed/altered by the server.
* WebApp: The webapp has two modes a. User mode and b. Admin mode.It shows the user specific data(like balance, user name) to the corresponding user and all the data to the admin.

# PROCESS FLOW CHART

# DATABASE DESIGN



# COMMAND FORMAT

**RFID Detection (R0)**

Lock → Server

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDR |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: R0 |
| RESERVE DATA | Reserve Data(0) |
| VALUE | User ID |
| VALUE | Unlock time stamp(Time stamp by second(s)，10 bytes) |
| END | Command end Code: #<LF> |

Example : \*CMDR,OM,863158022988725,1497689816,R0,0,246.104.222.116.52,1497689816#<LF>

**UNLOCK COMMAND (L0)**

Server → Lock

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDS |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: L0 |
| VALUE | Unlock(0) |
| VALUE | User ID |
| VALUE | Unlock time stamp(Time stamp by second(s)，10 bytes) |
| END | Command end Code: #<LF> |

Example: \*CMDS,OM,863158022988725,170619195455,L0,0,20,1497689816#<LF>

Lock →Server

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDR |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: L0 |
| VALUE | Unlock(0) |
| VALUE | User ID |
| VALUE | Unlock time stamp(Time stamp by second(s)，10 bytes) |
| END | Command end Code: #<LF> |

Example : \*CMDR,OM,863158022988725,1497689816,L0,0,246.104.222.116.52,1497689816#<LF>

Server Response

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDS |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: RE |
| VALUE | The server responds with the corresponding Command code L0 |
| END | Command end Code: #<LF> |

Example : \*CMDS,OM,863725031194523,001497689816,Re,L0#<LF>

**LOCK COMMAND (L1)**

Lock → Server

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDR |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: L1 |
| VALUE | User ID |
| VALUE | Unlock time stamp(Time stamp by second(s)，10 bytes) |
| END | Command end Code: #<LF> |

Example : CMDR,OM,863158022988725,1497689816,L1,246.104.222.116.52,1497689816#<LF>

Server Response

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDS |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: RE |
| VALUE | The server responds with the corresponding Command code L1 |
| END | Command end Code: #<LF> |

Example: \*CMDS,OM,863158022988725,000000000000,Re,L1#<LF>

**POSITION COMMAND(D0):**

Server → Lock

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDS |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: D0 |
| END | Command end Code: #<LF> |

Example : CMDS,OM,863725031194523,000000000000,D0#<LF>

Lock → Server

CMDR,OM,863158022988725,000000000000,D0,0,124458.00,2237.75314,11408.62621#<LF>

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDR |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: D0 |
| Reserve Data | 0 |
| VALUE | UTC Time |
| VALUE | LATITUDE |
| VALUE | LONGITUDE |
| END | Command end Code: #<LF> |

Example: CMDR,OM,863158022988725,000000000000,D0,0,124458.00,2237.75314,11408.62621#<LF>

Server Response

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDS |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: RE |
| VALUE | The server responds with the corresponding Command code D0 |
| END | Command end Code: #<LF> |

Example: CMDS, OM,863725031194523,000000000000,Re,D0#<LF>

**OBTAIN LOCK STATUS COMMAND (S5)**

Server → Lock

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDS |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: S5 |
| END | Command end Code: #<LF> |

Example : CMDS,OM,863725031194523,000000000000,S5#<LF>

Lock → Server

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDR |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: S5 |
| VALUE | Battery Volume |
| VALUE | GSM signal Value |
| VALUE | Reserve Parameter |
| VALUE | Lock Status(0- Unlocked, 1-Locked) |
| Value | Reserve Parameter |
| END | Command end Code: #<LF> |

Example : \*CMDR,OM,863158022988725,000000000000,S5,410,31,00,1,0#<LF>

**HALT COMMAND (L2)**

Lock →Server

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDR |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: L2 |
| VALUE | User ID |
| VALUE | Unlock time stamp(Time stamp by second(s)，10 bytes) |
| END | Command end Code: #<LF> |

Example: CMDR, OM, 863158022988725,00000000,L2,246.104.222.116.52,1497689816#<LF>

Server Response

|  |  |
| --- | --- |
| STX | Data header/frame header, fixed value: \*CMDS |
| CODE | Device Code |
| IMEI | Device(Lock) IMEI number |
| TIME | Local Time-Year/Month/Day/Hour/Min/Sec : YYMMDDHHMMSS |
| CMD | Command Code: RE |
| VALUE | The server responds with the corresponding Command code D0 |
| END | Command end Code: #<LF> |

Example: CMDS, OM,863725031194523,000000000000,Re,D0#<LF>

# PROCESS FLOW WITH COMMANDS

* **Scan RFID**(to start a ride)
* **Authentication command-R0 command**(lock to server)
* **Authentication at the server**
* **Unlock Commands-L0 command**(server to lock)-->(lock to server)-->(server response)
* **Periodic Position Commands-D0 command**(server to lock)-->(lock to server)-->(server response)
* **Rider Locks the Bicycle without pressing Push Button i.e. halting case -L2 command**(lock to server)-->(server response)
* **Scan RFID**(to continue the ride)
* **Lock unlocks after verifing the current rider.**
* **Status update in server**

(lock to server)-->(server response)

* **Rider locks the Bicycle after pressing Push Button i.e. end ride case**

**1. GPS location sent to server** (lock to server)-->(server response)

**2. Lock command sent to server- L1 command** (lock to server)-->(server to lock)  
**3. If not in Geo-fenced area**

**Unlock command sent to server-L0 command** (server to lock)-->(lock to server)-->(server response)BACKEND SERVER

1. Connects to the Database consisting of tables as shown in the class diagram above.
2. Registers a new user in the database.
3. Checks the credentials of the user trying to log in against information of existing users.
4. Parses the command received from the lock device and take respective actions:

* Command - R0: Server checks the user id against the database-if it’s valid then sends a response
* Command – L0: Adds a new ride with the corresponding user and lock details , updates the lock status, and number of cycles in dock.
* Command – L1: checks the location of the lock

If the location is in one of geofenced areas, the particular ride is stopped and balance gets deducted from user’s account and sends a response to server.

Else the ride is taken to be halted and a response is sent.

* Command –D0: Updates the latitude and longitude of the lock in the database and sends a response to the device.
* Command – L2: Updates the lock status to halt and sends a response.

1. Geofencing Algorithm  
   Information about all the docks are already entered in the dock table of database. For geofencing the cycles are checked to be in a rectangular region centered at the location coordinates of the respective dock. Riders are allowed to end the ride only such geofenced areas.