OPSC7311 Portfolio of evidence

Duran Moodley 13016335

Assignment 2 Question 2

Contents

[Introduction 2](#_Toc452919882)

[Program Specifications 2](#_Toc452919883)

[Application Description 2](#_Toc452919884)

[Functionality 2](#_Toc452919885)

[System Requirements 2](#_Toc452919886)

[Flow Chart 4](file:///C:\Users\Duran\Desktop\PoE%20-%2013016335\Assignment%202\Question%202\Documentation\Documentation.docx#_Toc452919887)

[UML Diagrams 13](#_Toc452919888)

[Class Diagrams 13](#_Toc452919889)

[Use Case Diagram 17](#_Toc452919890)

[Database Design 17](#_Toc452919891)

[ERD 17](#_Toc452919892)

[Table Design 18](#_Toc452919893)

[Help File 19](#_Toc452919894)

[Screenshots/Test Data 20](#_Toc452919895)

[Validation 24](#_Toc452919896)

[Lecturer Feedback & Corrections Made 25](#_Toc452919897)

[Enhancements 26](#_Toc452919898)

[Conclusion 26](#_Toc452919899)

# Introduction

This application is being developed to assist the needs of the supervisor in charge of managing Gilroc, a building containing a flats situated in the Durban CBD area. The supervisor of the building uses a manual file system to keep track of everyone who lives in each flat. If a new occupants moves into the building, the supervisor needs to create a new file and enter the relevant information. In addition, it very difficult and tedious to update any information. Searching for the required file can be very time consuming.

# Program Specifications

## Application Description

The supervisor would like an application that can do the above functions efficiently and accurately. The app should also contain a search feature. A report should be produced every month of all tenants living in the building. The application should display and retrieve the following information: number of occupants staying in the flat, names of each occupant, ID numbers, contact details, the flat number the occupant will be staying in and the date of arrival. An addition, the app should also allow the supervisor to record any incidents that occur in the building and that need attention. This includes a flood, a break in or burst pipe and many more. The supervisor should also be able to view a report of all incidents.

## Functionality

* Applies CRUD principals using database
* Using 2 tables
* Send sms
* Searches for a tenant
* Reading from a text file using the assets folder

## System Requirements

**Menu Screen**

*This will present the user with a list of options that they can select from. This includes: Add Tenant, Delete Tenant, Update/Search Tenant, Show Tenant, Add/Show Incidents, help, about us and Send Sms.*

**Add Tenant**

*This will allow the supervisor to enter in all information relating to the tenant. Details that are entered are: tenant Id, tenant name, tenant contact and emergency number, flat number and number of occupants staying. After all input has been validated, clicking on the send button will insert the data to the database.*

**Delete Tenant**

*If a tenant moves out of the building, the supervisor may delete that tenant out of the database. The supervisor must enter the id number of the tenant. If the id number exists, the tenant is deleted.*

**Update/Search Tenant**

*The supervisor must first search for the tenant using their id number. Tapping the search button will show all details relating to that tenant if he/she exists in the database. The supervisor can now change any details about that tenant such as contact number.*

**Show Tenant**

*This displays all information relating to all tenants. Only the name and id number of the tenant will be displayed. If the supervisor wants additional information about that tenant, he can simply select the tenant on the list and a dialog box will display all details about that individual tenant.*

**Add/Show Incident**

*There are occasions when something happens in the building that needs to be recorded. This operation will allow the supervisor to do just that. He will be able to record the date and time of that incident, followed by a description and the id number of the person who reported it. In addition, selecting the show incident button will display all incidents that have occurred in the building.*

**Help**

*This will show detailed information to the user if he/she needs assistance and guidance on how to us the application.*

**About us**

*Gives a description to user about details about the developer and other interesting details. This information is retrieved from a text file.*

**Send Sms**

*This allows the supervisor to send a sms to a tenant in the building.*



# Flow Chart



















# UML Diagrams

## Class Diagrams











## Use Case Diagram



## Database Design

### ERD



### Table Design

**Tbl\_Tenant**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id Number[PK] | Tenant Name | Number  Occupants | Contact  Number | Emergency  Number | Flat  Number |
| 14253647581 | Dan | 2 | 0313038587 | 0713558977 | 34 |
| 46367463634 | Sipho | 3 | 0313067890 | 0723848383 | 28 |
| 35636746463 | Sandy | 2 | 0314048928 | 0713828382 | 11 |
| 46364636465 | Muhammed | 1 | 0313038938 | 0728382832 | 12 |
| 34838488383 | Wandle | 1 | 0313039329 | 0713287273 | 30 |

**Tbl\_Incident**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \_id[PK] | Id  Tenant[FK] | Incident  Name | Date | Time | Incident Rating |
| 1 | 14253647581 | Burst pipe | 6/3/2016 | 04:00 | Minor Problem |
| 2 | 14253647581 | flooding | 14/3/2016 | 20:00 | Major Problem |
| 3 | 46364636465 | Blackout | 20/3/2016 | 19:00 | Minor Problem |
| 4 | 34838488383 | Blackout | 22/3/2016 | 15:00 | Major Problem |
| 5 | 34838488383 | Burst Pipe | 30/3/2016 | 12:00 | Major Problem |

# Help File

This application can insert, delete, update and display all tenant information.

This application can also insert and display all incidents in the building.

**Insert**

This is easily done by simply adding the relevant data in each field.

Please make sure that all fields are entered.

Id number must be 11 digits and Phone numbers must be 10 digits.

Error messages will be displayed if the above is not followed correctly

**Update**

You need to first search for the Tenant by entering in their id number.

If not found no data will be shown.

Once the Tenant has been found, you may now update the relevant information.

Please follow the same processes as in the Insert section.

**Delete**

The Tenant's id number is required for this operation to be executed.

If the id number is not in the database, an error will be shown.

Please make sure that the right id number is entered.

**Display**

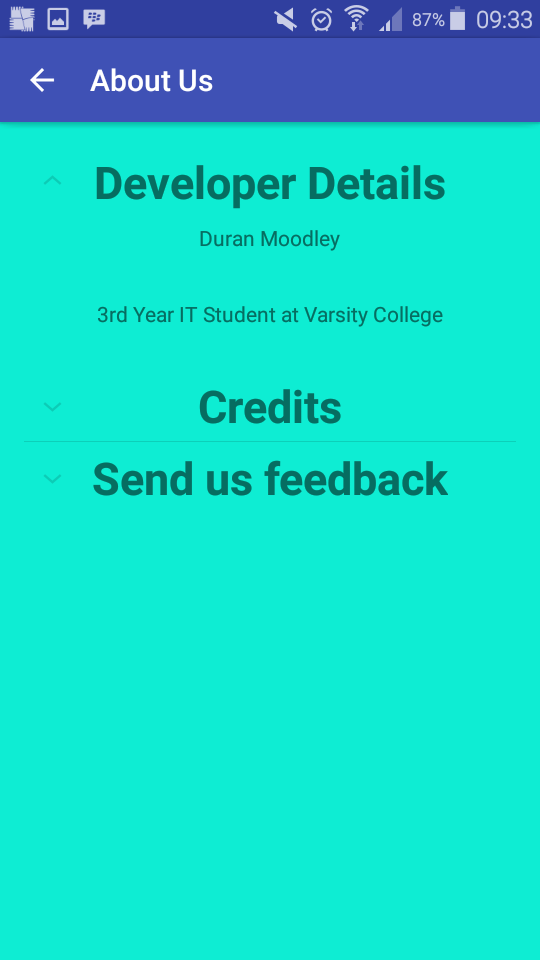
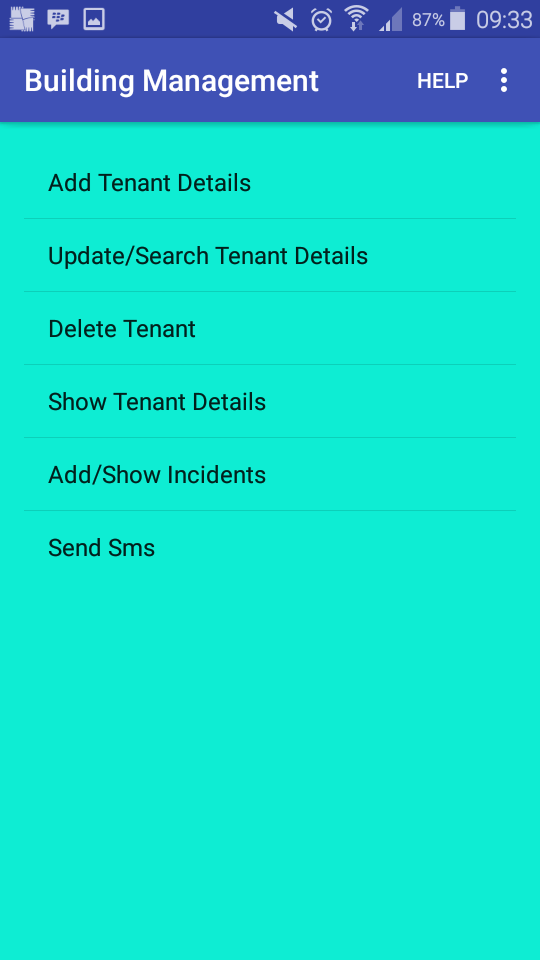
All the information will be shown in a list.

# Screenshots/Test Data

**Menu Screen** **About Us**

Presents the user with a set of options that Gives the user additional information

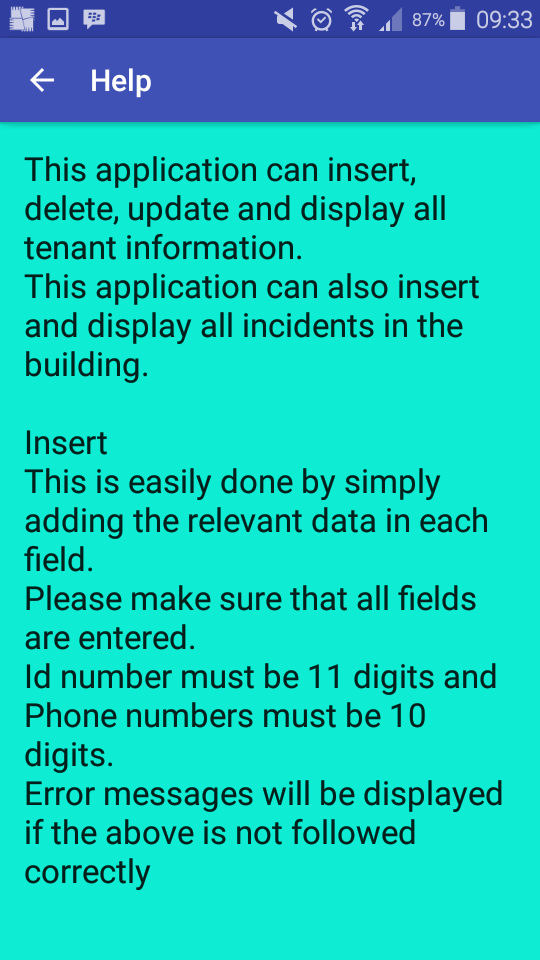
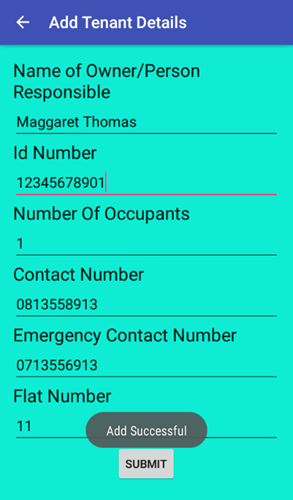
He/she can choose from about the application



**Help Add Tenant Details**

Gives the user guidance on how to use Allows the user to enter a new tenant

the application by tapping the Submit button



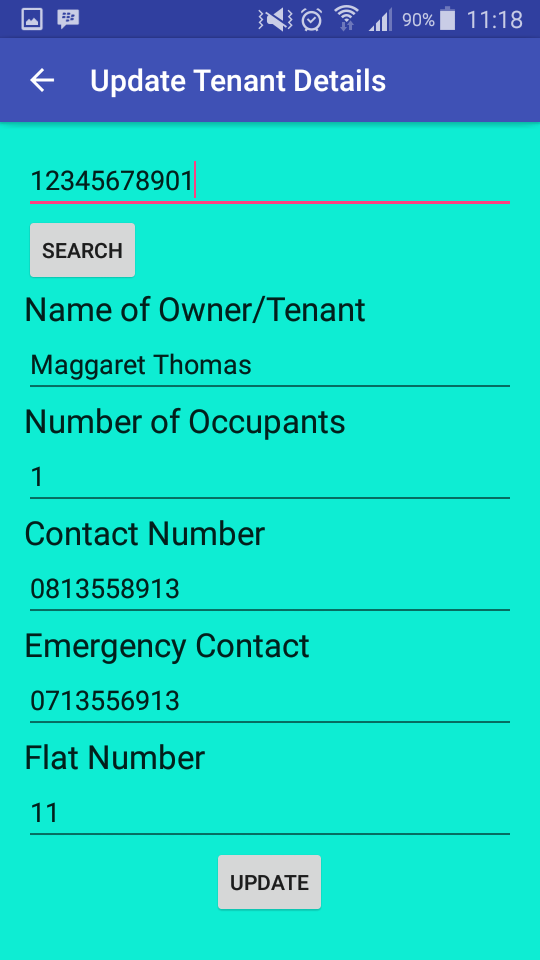
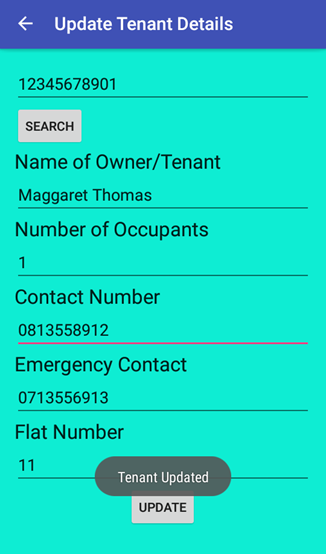
**Update Tenant** **Update Tenant**

Entering a tenant id and tapping search After making a change, the user

Will display all tenant details as seen below can tap the update button to register the

Change

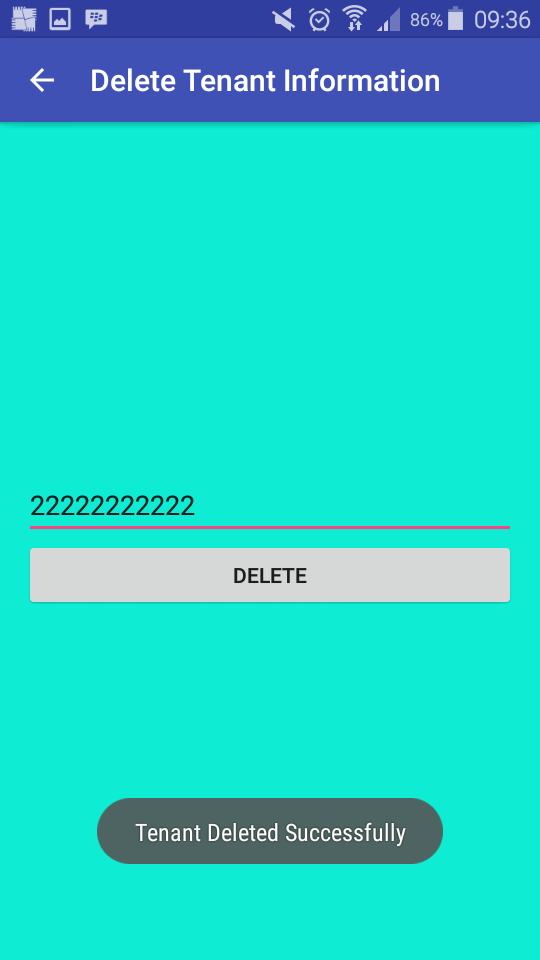
(Contact Number was changed)



**Delete Tenant**

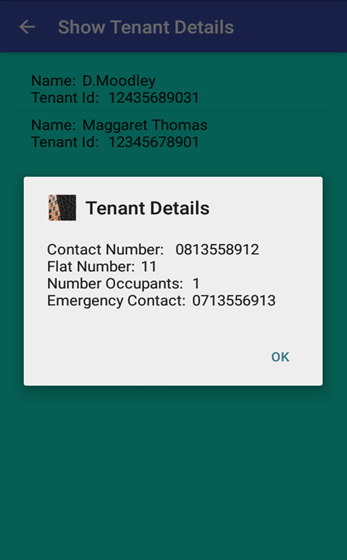
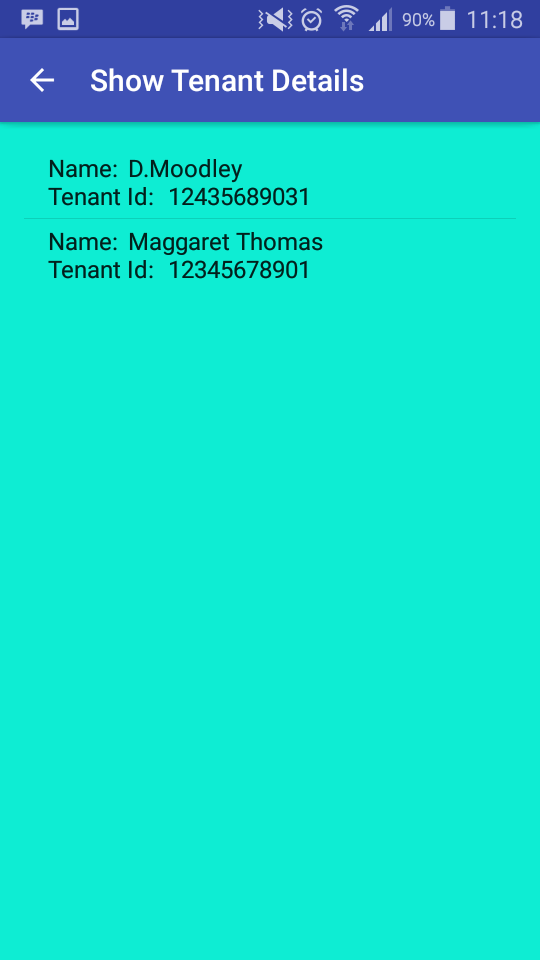
User must enter id number of tenant,

Selecting the delete button will remove the tenant



**Show Tenant Show Tenant**

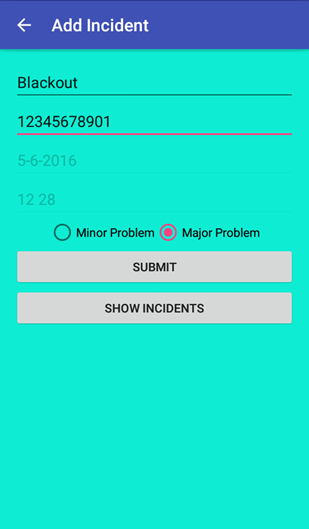
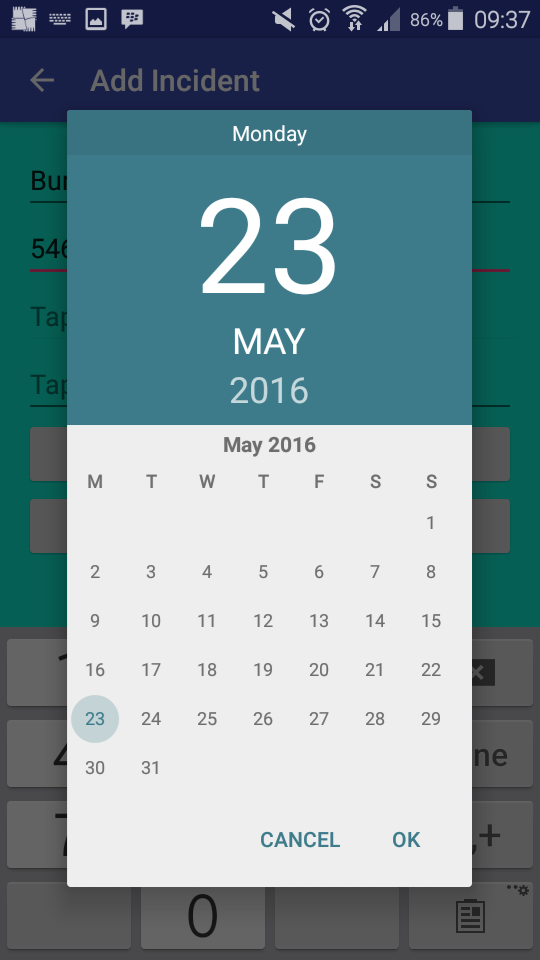
Displays a summarized version of all tenants Selecting a tenant from the list will display a pop up

 With detailed information about that tenant

**Add Incident Add Incident**

Allows the user to add an incident The user will need to enter in

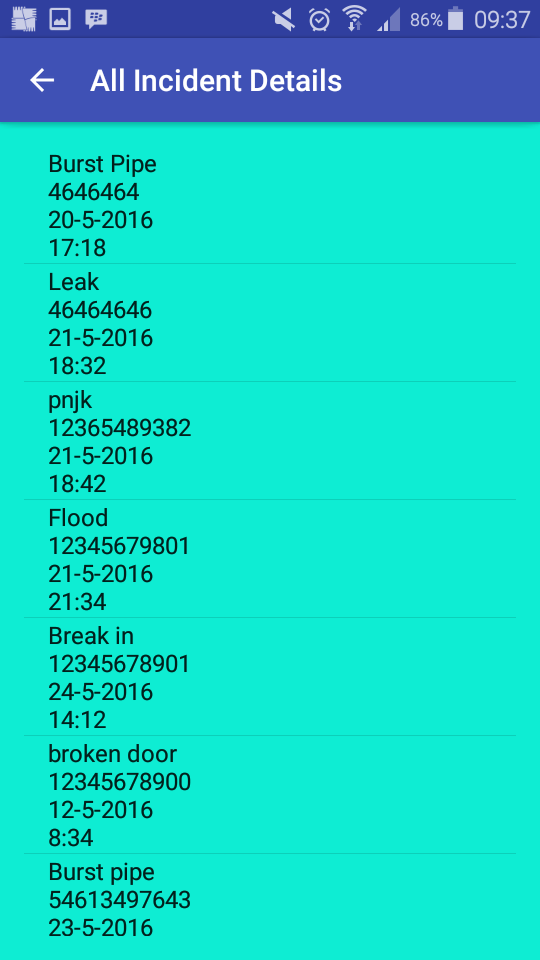
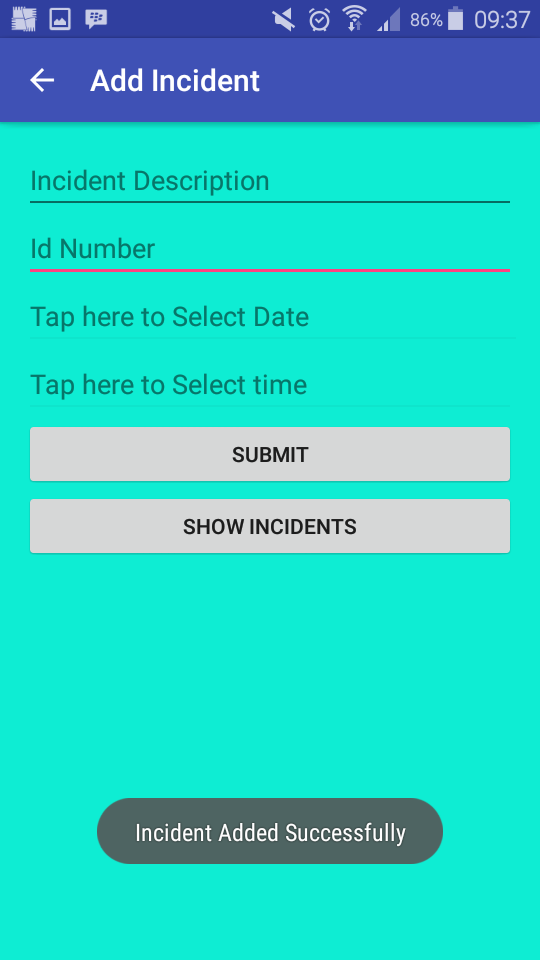
Date and time by using this date time picker



**Add Incident Show Incident Details**

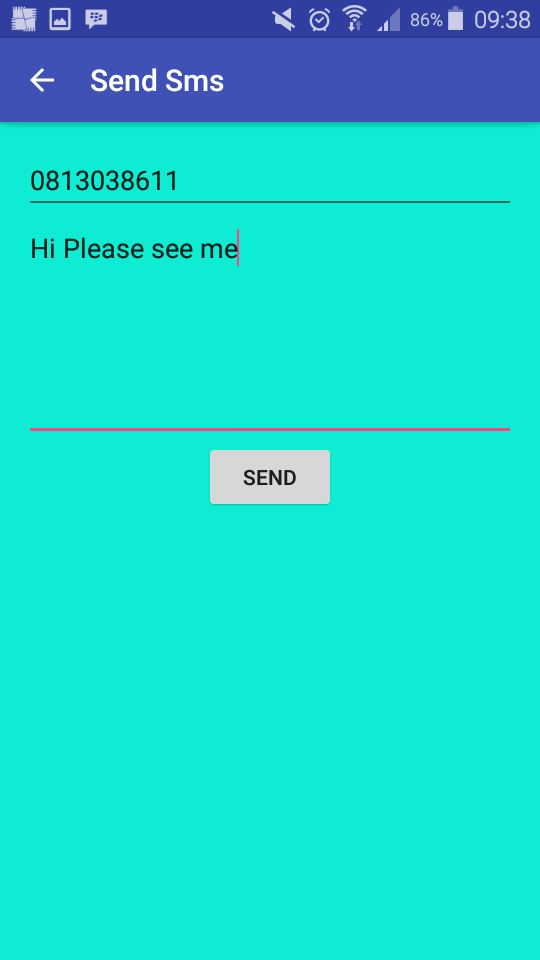
Tapping the submit button Shows all incidents to the user

Will add that incident



**Send Sms**

The user can send a sms by adding the relevant details



## 

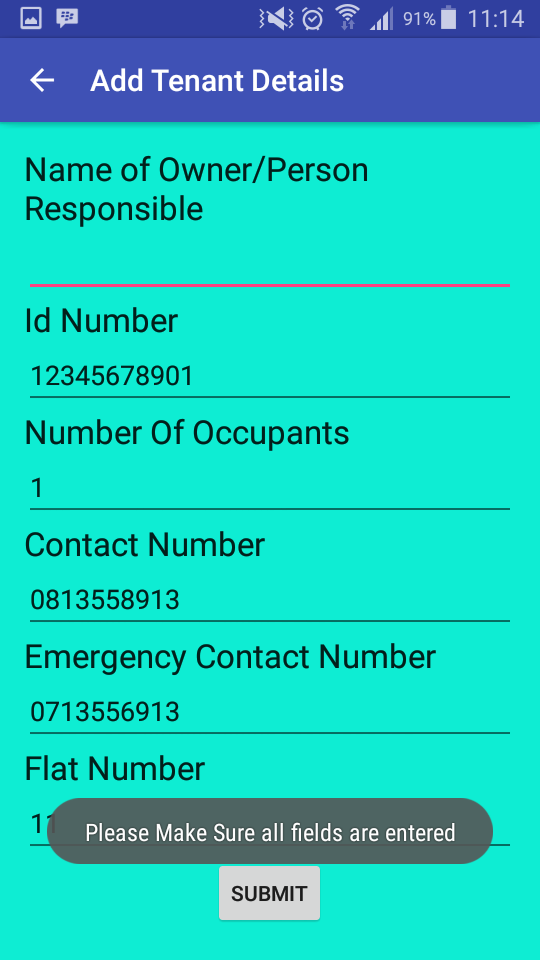
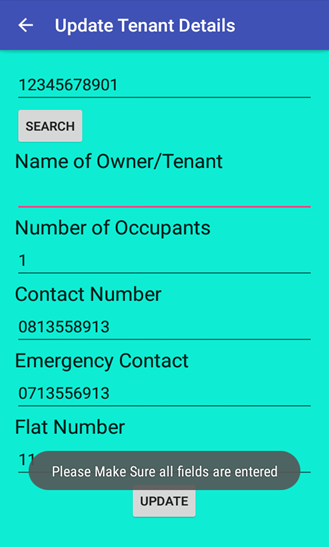
## Validation

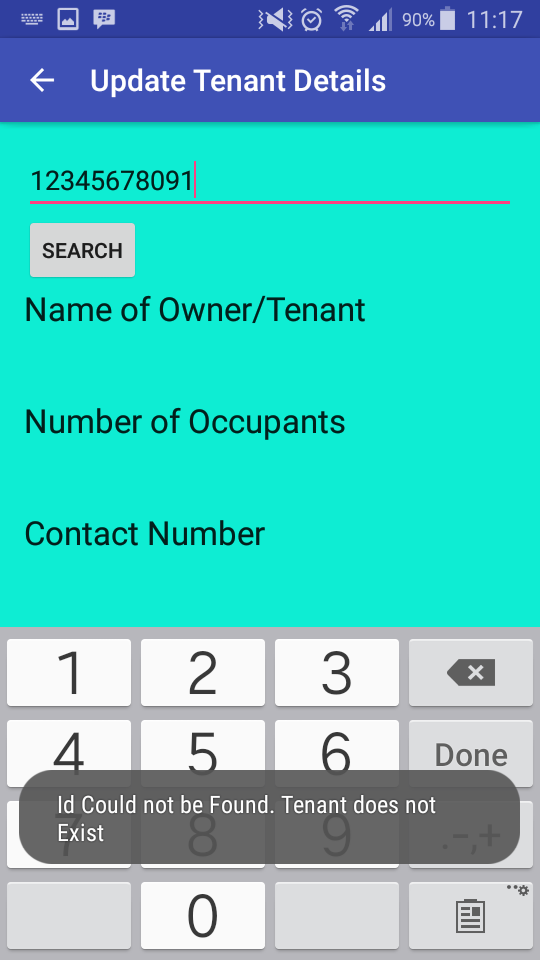
In this application I have made sure to incorporate extensive error check/validation. If a user makes a mistake when entering a field. The application will produce an error message which will give details on how to fix the error. The following screenshots displays some error messages that can occur.

**Add Tenant Error Message Update Tenant Error Message**

All fields must contain values.

The following message will be produced if this is violated





**Update Tenant Details Error Message**

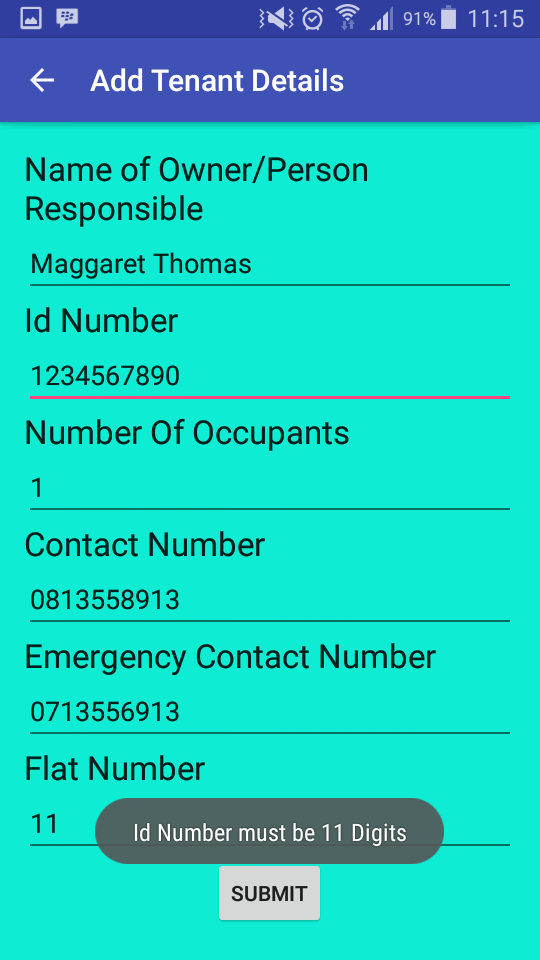
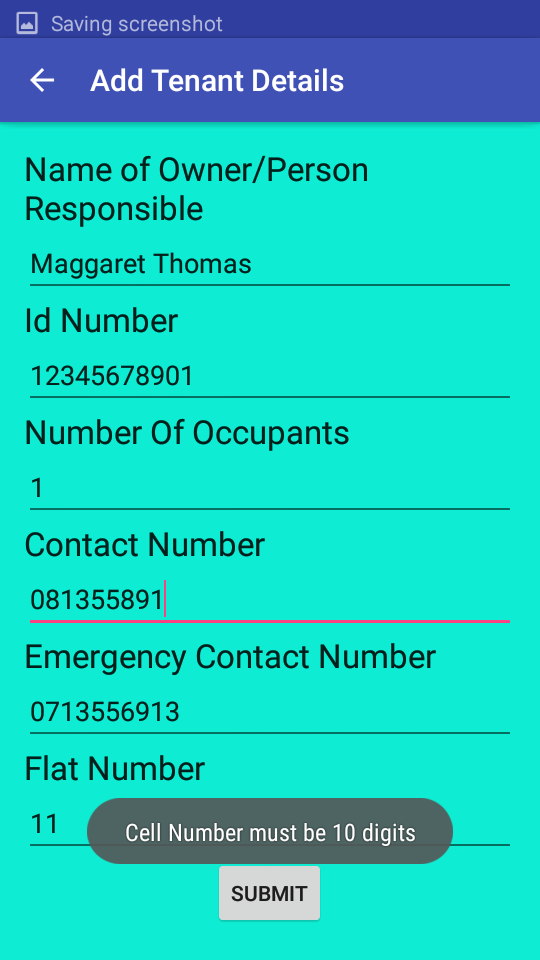
Entering an incorrect tenant id number to search for will produce an error.

**Add Tenant Error Message**

Incorrectly entering an incorrect number of digits

For an id number or contact number

will produce an error



# Lecturer Feedback & Corrections Made

|  |  |
| --- | --- |
| Lecturer Feedback | Corrections Made |
| Missing density folder in drawable | Folder is seen in the mipmap directory |
| Edit text date and time – input type must be specified | Input type specified |
| App not index able by google search | Not corrected |
| Painting region more than once | Not corrected |
| Don’t concatenate in set text | Concatenation from set text is removed |
| 2 unused resources | Unused resources removed |
| Use start/end instead of left/right | Start/end used |
| 3 unchecked warnings | Not corrected |
| 4 unused imports | Unused Imports removed |
| 24 items have potential of producing null errors | Assert and if statements added to check for null values |
| Use methods to remove redundancy | Methods created and called to remove redundancy |
| Tables not linked by foreign key | Tables are now linked |

# Enhancements

* Validation methods in update and insert tenant was changed to improve error checking.
* An additional method was created to display messages to the user. This method removes redundancy.
* Added additional strings to the strings.xml file and called those strings when an error is caught due to a mistake from the user.
* Added an additional field to the database. This field allows the supervisor to rate how big the incident or problem is in the building ie. Major or Minor problem.
  + An additional parameter was added to the insert Incident method to insert that field into the database.
  + This field was also added to the get all incident method.
  + These changes were made in the database adapter class.
* A radio group containing radio buttons was added to the add incident activity to accommodate for the retrieval of the newly added field in the database.
* Made use of the colors. Xml file by adding a background colour and calling it in each activity.
* Labels were added above all edit texts.

# Conclusion

This application has increased my knowledge with sql lite database. I am now able to store large amounts of data efficiently. In addition, I can also add, insert, delete and search for data in the database with ease. Later on, I hope to improve this app by adding more search features to filter large amounts of data.