Test Plan
Application BMS

Table of Contents

1 Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.A test case is the individual unit of testing. It checks for a specific response to a particular set of inputs and unittest provides a base class, TestCase, which may be used to create new test cases.

1.1 Objectives

The application is java based therefore we use the same for the testing purpose of our application. Java with gradle provides a small set of tools that come in handy when writing tests. To use the test client, instantiate `org.junit.Test`

. The test team is responsible for testing the product and ensuring it meets their needs. The test team is both the customer and the tester in this project.

Phase 1 of the project will test all the unit test cases and deliver the application for further advanced testings. This will allow the test team to start transferring tests over to the new system. Must have functionality is considered more important than the delivery date in this project.

2 Scope

The initial phase will include all 'must have' requirements. These and any other requirements that get included must all be tested. At the end of Phase 1, a tester must be able to:

1. Create a manual test and automated unit test with as many steps as necessary
2. Save it
3. Retrieve it and have the ability to view it when running the test
4. View results and appropriate comments

As the team works with the product they will define the needs for the second phase.

Load testing will not be considered part of this project since the user base is known and not an issue.

Rewriting, moving or porting existing test cases is not considered part of this project.

3 Modules /Test Cases

Test Cases	**Description of test cases**
Module - User	
Authorized login	Verify userto login only when credentials are valid.**(TC-1)**
Logout	On click Logout user is logged out.**(TC-7)**

Register	Create a new user with entered details and verify the details in the database.**(TC-2)**

Module - entry check

Choose building and room	Allows user to choose his required building and room **(TC-3)**
Display entry permission	Shows user if he is allowed to enter **(TC-4)**

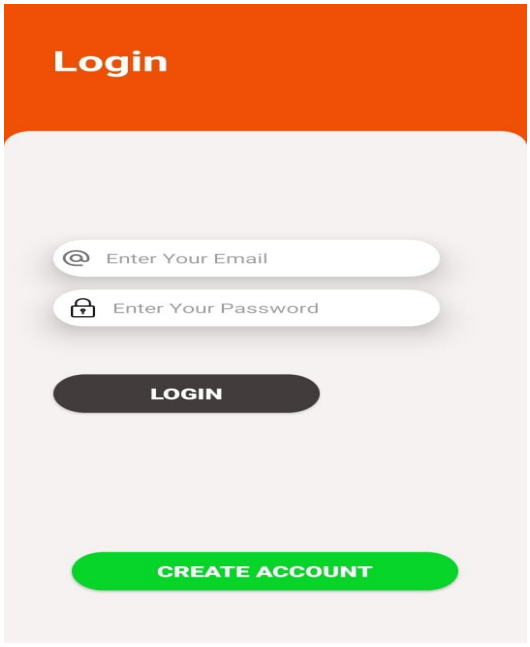
Module-admin monitoring

Admin login	Authorised admin login with his credentials **(TC-5)**
Occupancy monitor	Allows admin to check no of users in a room **(TC-6)**

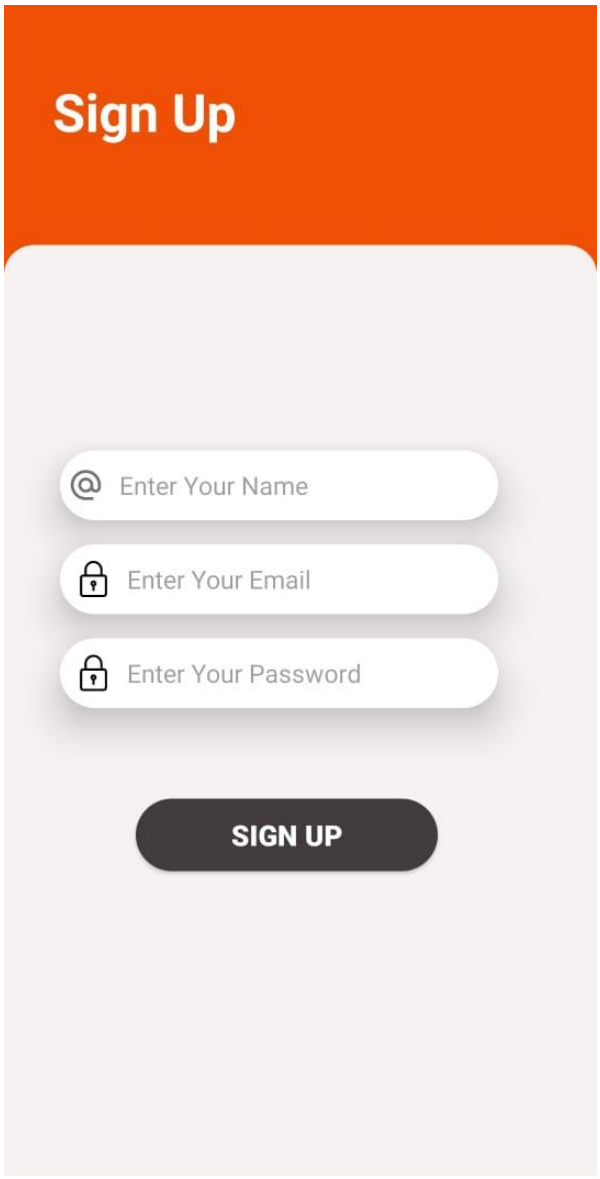
Logout	On click Logout user is logged out.**(TC-7)**

TEST CASE 1

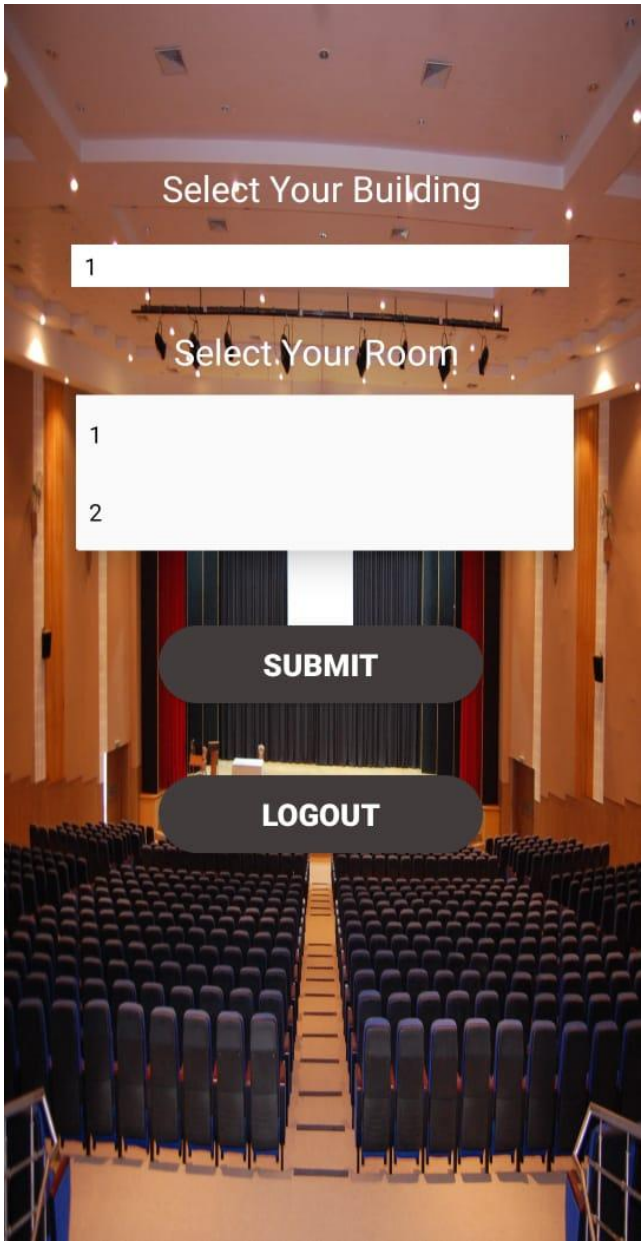
User login



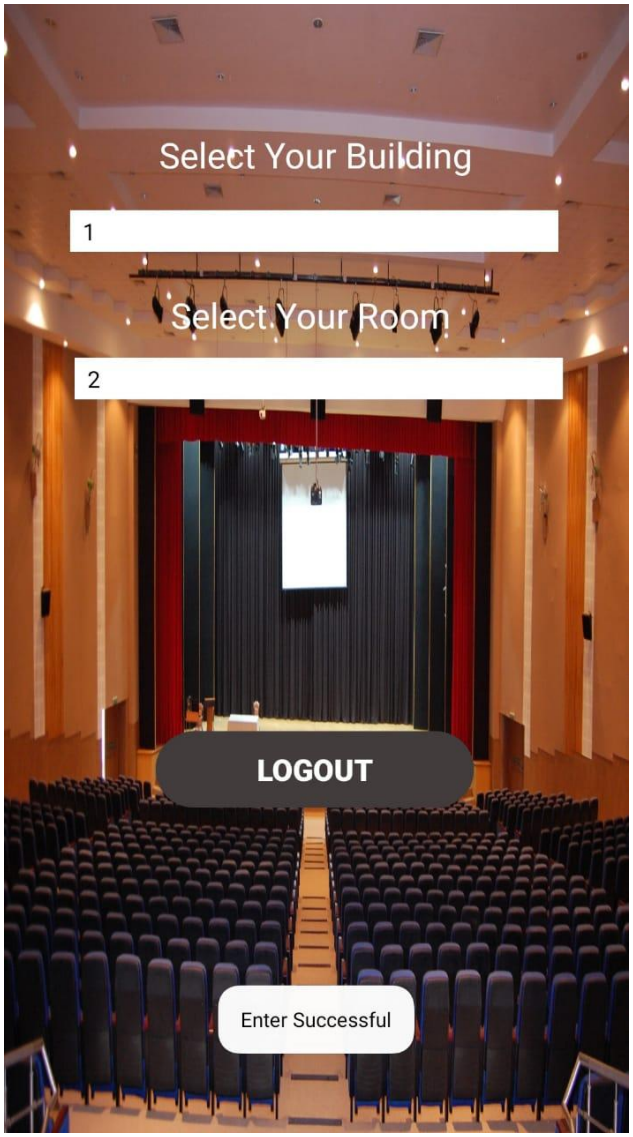
TEST CASE 2
REGISTER



TEST CASER 3
CHOOSE BUILDING AND ROOM



TEST CASE 4
CHECK OCCUPANCY



Test case 5
Occupancy monitoring


00:15
Room1: 55
Room2: 33

4 Assumptions / Risks

4.1 Assumptions

This section lists assumptions that are made specific to this project.

1. The application is fully functional and bug free to the best of the developer team's knowledge and ability.
2. Web-app must be completely responsive and must run smoothly on all the devices whether it may be laptops,desktops or mobile phones.
3. Sensitive files and data of the application must be completely secured and must not be accessible to any individual.

4.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

#	Risk	Impact Trigger	Mitigation Plan
1	Scope Creep – as testers become more familiar with the tool, they will want more functionality	High Delays in implementation date	Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders. Since the driver is functionality and not time, it may be necessary to push the date out.

2	Changes to the functionality may negate the tests already written and we may lose test cases already written	High – to Loss of all schedule test cases and quality	Export data prior to any upgrade, massage as necessary and re-import after upgrade.f
3	Weekly delivery is not possible because the developer works off site	Medium Product did not get delivered on schedule	Weekly tasks must be completed on time.

5 Test Environment

A new server is required for the app server, the application and the database which is taken care of by the firebase app and we only need to worry about the testing part. Separate unit tests cases must be made to avoid false judgements due to previously fine running tests.

6 Hardware Requirements

- Operating System: android 8
- Hard Disk: 8gb
- RAM: 256 MB
- Processor:

7 Test Approach

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never used this type of

tool and will be learning as they go. Tests for planned functionality will be created and added to TCT as we get iterations of the product.

7.1 Test Automation

Automated unit tests are part of the development process, and images for the same have been provided below

```java
package com.nik.projectclass;

import android.content.Context;

import androidx.test.platform.app.InstrumentationRegistry;
import androidx.test.ext.junit.runners.AndroidJUnit4;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

/**
 * Instrumented test, which will execute on an Android device.
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Test
    public void useAppContext() {
        // Context of the app under test.
        Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
        assertEquals("com.nik.projectclass", appContext.getPackageName());
    }
}
```

Module Testing-

8 Further Testing Approach

Unit Testing Completed
System and Integration Testing
Performance Testing
Acceptance Testing
Configuration Testing