Rental Property Management: An Android Application

GRADUATE PROJECT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Prateek Shivraj
Spring 2015

Committee Members

Dr. Ajay Katangur
Committee Chairperson

Dr. Dulal Kar
Committee Member
ABSTRACT

A majority of unpleasant issues faced by landlords and tenants seem to grow due to lack of communication. Late rent payments turn into an eviction. A non-functional AC or a broken window becomes a reason to break a lease. Fortunately, building a positive relationship doesn't take much effort and it all starts with one basic idea: communication. Rental Property Management app provides the solution by offering the ability to share issues between tenants and landlords which leads to rapid issue resolution and fewer misunderstandings. The goal of the app is to create a better relationship between tenants and landlord. The Rental Property Management app is designed to support both tenants and landlords by enabling them to document and communicate repair issues, send automatic rent reminders, package notifications and emergency information. It helps the tenant with making Payments and it also helps the Landlord to keep track of issues posted by tenants. Effectively resolving the apartment issues is important to the tenant's long-term future and the Rental Property Management app will be an important tool for creating rental housing stability by helping tenants speak with greater credibility through initiating and documenting communications and building productive relationships with landlords.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>1 Background and Rationale</td>
<td></td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Android Studio</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Parse</td>
<td>4</td>
</tr>
<tr>
<td>1.3.1 Cloud Code</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Existing App</td>
<td>5</td>
</tr>
<tr>
<td>1.5 Proposed App</td>
<td>5</td>
</tr>
<tr>
<td>2 NARRATIVE</td>
<td></td>
</tr>
<tr>
<td>2.1 Problem Statement</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Motivation</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Product Description</td>
<td>8</td>
</tr>
<tr>
<td>2.4 Product Scope</td>
<td>8</td>
</tr>
<tr>
<td>3 Product Design</td>
<td></td>
</tr>
<tr>
<td>3.1 Product Design and Architecture</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Use Case Diagram</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Class Diagram</td>
<td>11</td>
</tr>
<tr>
<td>3.4 Sequence Diagram</td>
<td>12</td>
</tr>
<tr>
<td>3.4.1 Report an Issue</td>
<td>12</td>
</tr>
</tbody>
</table>
3.4.2 Rent Tab .......................................................................................................................... 13
3.4.3 Package Tab ...................................................................................................................... 15
3.4.4 Utilities ................................................................................................................................ 16
3.4.5 Emergency Information ..................................................................................................... 17
3.4.6 Documents .......................................................................................................................... 18
3.4.7 Logout .................................................................................................................................... 19
3.4.8 Add Apartment .................................................................................................................... 20

4 SYSTEM IMPLEMENTATION ................................................................................................. 21
4.1 Rental Property Management ............................................................................................... 21
4.2 Landlord .................................................................................................................................. 23
   4.2.1 Update Apartment .............................................................................................................. 24
   4.2.2 Navigation Drawer (landlord) ............................................................................................ 25
4.3 Tenant ...................................................................................................................................... 26
   4.3.1 Navigation Drawer (tenant) ............................................................................................... 27
   4.3.2 Report an issue form .......................................................................................................... 28
   4.3.3 Rent Payments .................................................................................................................... 29
4.4 Push Notification ..................................................................................................................... 30
4.5 Cloud Server ........................................................................................................................... 31
4.6 Cloud Code ............................................................................................................................... 32

5 EVALUATION AND TESTING ............................................................................................... 33
5.1 Registration .............................................................................................................................. 33
5.2 Login ......................................................................................................................................... 35
5.3 Forgot Password ....................................................................................................................... 36
5.4 Floating Icon in Issues Page .................................................................................................... 38
5.5 Add Apartment .......................................................................................................................... 40
LIST OF FIGURES

Figure 1.1 Intelligent Code Editor ................................................................. 3
Figure 1.2 Multiple screen app development ................................................. 4
Figure 3.1 Rental property management Architecture ................................. 9
Figure 3.2 Use case Diagram ...................................................................... 11
Figure 3.3 Class diagram .......................................................................... 12
Figure 3.4 Sequence diagram for reporting issue ......................................... 13
Figure 3.5 Sequence Diagram for Rent Payment ........................................... 14
Figure 3.6 Sequence Diagram for Package Notifications ............................ 15
Figure 3.7 Sequence Diagram for Utilities .................................................. 16
Figure 3.8 Sequence Diagram for Emergency Information ........................... 17
Figure 3.9 Sequence Diagram for Documents ............................................ 18
Figure 3.10 Sequence Diagram for Logout .................................................. 19
Figure 3.11 Sequence Diagram for Apartment .......................................... 20
Figure 4.1 Login and Registration ............................................................... 21
Figure 4.2 Landlord Home Screen ............................................................... 22
Figure 4.3 Landlord Home Screen (sorted list) ............................................. 23
Figure 4.4 Update Apartment & List view ................................................... 24
Figure 4.5 Navigation Drawer ................................................................. 25
Figure 4.6 Tenant Home Screen ............................................................... 26
Figure 4.7 List view in Navigation Drawer ................................................................. 27
Figure 4.8  Report an issue form .................................................................................. 28
Figure 4.9 Payments ..................................................................................................... 29
Figure 4.10 Push Notification ....................................................................................... 30
Figure 4.11 Snippet to connect with parse database .................................................. 31
Figure 4.12 Snippet to send Push Notifications .......................................................... 32
Figure 5.1 User Sign up Page ....................................................................................... 34
Figure 5.2 User Login Page ......................................................................................... 35
Figure 5.3 Reset Password ......................................................................................... 36
Figure 5.4 Password link to inbox ............................................................................. 37
Figure 5.5 Floating Icon ............................................................................................. 38
Figure 5.6 Sort Issues ................................................................................................ 39
Figure 5.7 Add Apartment ......................................................................................... 40
Figure 5.8 List of Apartments .................................................................................... 41
Figure 5.9 Sort List of Apartments ............................................................................. 42
Figure 5.10 Apartment Details .................................................................................. 43
Figure 5.11 Package List ............................................................................................ 44
Figure 5.12 Issue Details ........................................................................................... 45
Figure 5.13 Update Status ......................................................................................... 46
Figure 5.14 Zoom Picture .......................................................................................... 47
Figure 5.15 This month Rent .................................................................................... 48
Figure 5.16 Update Rent ............................................................................................ 49
Figure 5.17 Package Notifications .......................................................................... 50
Figure 5.18 Utilities bill .............................................................................................. 51
1 Background and Rationale

1.1 Introduction

Mobile Application Software have emerged in the past and are a hot trend in the market. The most common mobile operating systems are: Android from Google, iOS from Apple, Blackberry and Windows Phone from Microsoft. Android is free and open source mobile platform based on Linux kernel for developing apps from Google. Android is primarily designed for touch screens mobile devices. Android 5.0 "Lollipop" is the latest version of android operating system which features design changes in user interface built around a design language called material design. Following are a list of Android Operating System releases:

1. Cupcake (1.5)
2. Donut (1.6)
3. Éclair (2.0)
4. Frozen Yogurt (Froyo) (2.2)
5. Ginger Bread (2.3)
6. Honeycomb (3.0)
7. Ice Cream Sandwich (4.0)
8. Jelly Bean (4.1,2,3)
9. Kit Kat (4.4)
10. Lollipop (5.1)
Android has gained market share over the recent years over iOS, Windows, Blackberry, etc., [1]. The main reason being its open source and has been picked by many device manufacturers as the OS of their choice [2]. As android is popular, this project focuses on developing a mobile application, Rental Property Management for Google’s Android operating system. This project is developed for Android based smart phones and tablets on Android Studio, an IDE for developing applications on the Android platform.

1.2 Android Studio

Android Studio is an Integrated Development Environment (IDE) used for developing apps on the android platform. The new SDK packages and add-on API has made the IDE more interactive and easy to use. The material design helps to build an attractive UI. It is built on the popular IntelliJ IDEA (Community Edition) Java IDE [3]. Some of the important functionalities are:

Sample Importing & templates: Android Studio includes wizards that enables to start with new project templates or import Google code samples.

Code Editing: Android Studio takes advantage of all the intelligent code editing capabilities of IntelliJ IDEA such as advanced code completion, refactoring, and code analysis as shown in Figure 1.1 [4].

Internationalization string editing: It manages string translations of the app in Android Studio.
**Figure 1.1 Intelligent Code Editor**

*User interface design:* It lets users edit and preview Android Layouts across multiple screen sizes, languages, and even API versions as shown in Figure 1.2.

*Memory monitor:* It lets you view the memory usage of an app over time to help find ways to improve its performance.
1.3 Parse

Parse is a backend cloud storage which provides a way to link the mobile apps to access data, send push notifications, integrate with social networking sites and supports the ability to add rich custom logic to the app's backend with cloud code [5].

1.3.1 Cloud Code

Parse cloud code helps to build mobile apps without dealing with other servers. For complex apps, sometimes you just need a bit of logic that isn't running on a mobile device. Cloud code makes this possible. Cloud code is built on JavaScript SDK that powers many apps. The only difference is that this code runs in the Parse cloud rather than running on a mobile device. When cloud code is updated, it becomes available to all
mobile environments instantly. This lets you change app behavior on the fly and add new features faster [5].

1.4 Existing App

Squared way Chicago is a tenant/landlord web app which is designed to support both tenant and landlord, it allows to

- Communicate repair issues
- Collect data and inform about the trends in rental housing market

It is a web app which can be accessed at http://www.squaredawaychicago.com. The app development is funded through a grant from the MacArthur Foundation and support provided by Marguerite Casey Foundation and Chicago Community Trust [6].

Disadvantages of the existing app

- It must be accessed through a web browser.
- It is not a mobile app.
- It cannot send push notifications to the user.
- The issues are sent to the landlord's email and communication can only be achieved through messaging/email.
- Only restricted to Chicago city.

1.5 Proposed App

Rental Property Management app is a mobile application which is designed to support both landlord/tenant where
• The tenants can report repair or maintenance issues. They also have an option to take a picture of the issue and upload, which creates a visual record accessible to the landlord instantly.

• The tenants can pay the bill through the app.

• The tenants can keep track of all the notifications from the landlord as rent reminders, package notifications, and emergency info.

• The landlord can send package notifications, emergency info, rent reminders, utilities bill and documents to tenants.

• The app provides landlord/tenant to store important documents.

**Advantages** of proposed app

• Communication is through in-app messages.

• It can send push notifications

• It is convenient and easy to use.

The mobile app will help the landlord and tenant to overcome hassles and manage tasks and issues easily in a more convenient way.
2 NARRATIVE

2.1 Problem Statement

In our day-to-day life, tenant’s face a lot of issues related to household such as reporting an issue, tracking package in leasing office and receiving emergency information instantly etc., and similarly landlord’s also deal with issues such as keeping track of issues in apartments, notifying that a package has arrived, sending emergency information quickly etc. Currently there is no app which focuses on to resolve this particular issue. The existing web app Squared away as discussed in section 1.4 does help to solve the issue but not in an effective manner and is only limited to Chicago city. To help overcome the problems faced by tenants and landlords, a new app is developed. This app focuses on to build a better relationship between tenant and landlord by simplifying many tasks such as sending automatic rent reminders, package notifications, utilities bill, emergency info, documents and sharing issues.

2.2 Motivation

The tenant may not be able to visit or call the leasing office every time when an issue arises and at times it will be hard for the user to even clearly communicate the problem. Maintaining a record of issues from tenants is also a tedious process for the Landlord. Since most users use mobile phones now-a-days, an app is developed which makes this process simple by allowing the tenant to report the problem through the app. The app requires the tenant to fill out the form, take a picture and send it to the landlord.
All the issues are stored in a database which can be viewed by the landlord through the app.

The landlord will receive the issue in a well-documented form, which gives landlord a perspective on how to deal with the problem. In this way, Rental property management helps both tenant and landlord to build a better relationship by resolving issues swiftly.

2.3 Product Description

In order to use the product both tenant and landlord should to register with in the app, tenant's login details will be created by the landlord and the tenant can login in to the app with the details provided. The tenant can report an issue, pay the rent and utilities bill through the app which is displayed to the landlord. The landlord can update the rent and utilities bill, send emergency info and store the documents on cloud which can be viewed by tenant.

2.4 Product Scope

This app is designed for android users and works on both a phone and a tablet. The users of the product can use the services of the Rental property management when they are connected to the internet as the data exchange is from the parse cloud server. The product is compatible with different versions of android, starting from Android 3.0(Honeycomb) to the most recent update Android 5.1 (Lollipop). The app is designed with the latest features provided in Android 5.1 which is material design.
3 Product Design

3.1 Product Design and Architecture

Figure 3.1 shows the system architecture which illustrates the way, the communication happens between landlord and tenant’s. The user installs the app on the Android phone and both the landlord and tenant are navigated to their respective screen based on their login information.

![Figure 3.1 Rental property management Architecture](image)

Figure 3.1 Rental property management Architecture
The data exchange between landlord and tenant is achieved through parse database and the application data stored on parse cloud. The parse cloud server acts as an intermediate for the interaction between the landlord and tenant's.

### 3.2 Use Case Diagram

In the use case diagram shown in Figure 3.2, the use cases represent the functionalities of the landlord and tenant. Apart from registration, login and storing documents which is common between tenant and landlord, landlord will be updating rent, utilities, emergency info which is presented on the tenant's user interface. Tenant will have functionalities such as reporting issue's and bill payment which is displayed on landlord's screen.
3.3 Class Diagram

Class diagram consists of the interfaces, methods, variables and relationship between them. Figure 3.3 is a class diagram for the Rental Property Management app which describes the major functionalities of the landlord-tenant interface like registering with the application, login authentication etc. The functionalities of landlord are add apartment, update rent, sending package notifications, adding utilities and garbage bill, sending emergency info to all tenants and uploading lease documents. Both registration and login are the common classes for landlord and tenant but the screens and functionalities vary based on the attribute role. Similarly, the functionalities of tenants are to report issues and bill payments.
3.4 Sequence Diagram

A Sequence diagram describes how the communication happens between the user, application, and cloud data. Figure 3.4 describes the user sequence of handling a user request for reporting an issue. The application processes the user request and makes a request to the Parse server to get the data from the cloud database.

3.4.1 Report an Issue

Figure 3.4 is a sequence diagram for reporting issues which shows the functionality of tenant reporting an issue, which gets updated in parse cloud database and gives a view of the data to the landlord.
Figure 3.4 Sequence diagram for reporting issue

3.4.2 Rent Tab

The Figure 3.5 is a sequence diagram for Rent Payment which shows the functionality of landlord updating the rent, which gets updated in parse cloud database and gives a view of the data to the tenant. The tenant can pay the rent which is then notified to landlord.
Figure 3.5 Sequence Diagram for Rent Payment
3.4.3 Package Tab

Figure 3.6 is a sequence diagram for package notifications which shows the functionality of landlord sending package notification to the tenant, which gets updated in parse cloud database and gives a push notification to the tenant.

Figure 3.6 Sequence Diagram for Package Notifications
3.4.4 Utilities

Figure 3.7 is a sequence diagram for utilities which shows the functionality of landlord updating the Garbage and Water bill which gets updated in parse cloud database and gives a view of the data to the tenant. The tenant can pay the bill which is then notified to the landlord.

![Sequence Diagram for Utilities](image)

Figure 3.7 Sequence Diagram for Utilities
3.4.5 Emergency Information

Figure 3.8 is a sequence diagram for emergency information which shows the functionality of landlord sending Emergency Information to all the tenants, which gets updated in parse cloud database and gives a view of the data to the tenant. The tenant will also receive a push notification.

![Sequence Diagram for Emergency Information](image)

Figure 3.8 Sequence Diagram for Emergency Information
3.4.6 Documents

Figure 3.9 is a sequence diagram for documents which shows the functionality of the landlord updating image or text files, which gets updated in parse cloud database and gives a view of the data to the tenant. The tenant can view the documents.

![Sequence Diagram for Documents](image)

**Figure 3.9 Sequence Diagram for Documents**
3.4.7 Logout

Figure 3.10 is a sequence diagram for logout which terminates the current session of a user.

![Sequence Diagram for Logout](image)

Figure 3.10 Sequence Diagram for Logout
3.4.8 Add Apartment

Figure 3.11 is a sequence diagram for apartment which shows the functionality of landlord adding apartments, which gets updated in parse cloud database and gives a updated view to the landlord.

![Sequence Diagram for Apartment](image)

Figure 3.11 Sequence Diagram for Apartment
4  SYSTEM IMPLEMENTATION

The mobile app Rental Property Management is developed for Android using Android Studio, the web service is implemented in Parse and the user data is stored in parse cloud database in the backend. Push Notifications are implemented in Parse cloud code using Parse Window console. The rest of the chapter talks about the implementation details.

4.1  Rental Property Management

The Rental Property Management is developed using Android Studio 1.1 on Windows 8.1. The application is designed with a simple user interface with the purpose of providing great usability to the users. After downloading and installing the application the user has to register with the application using his identification information (username, password, email and Landlord/Tenant) as shown in Figure 4.1.

Figure 4.1 Login and Registration

21
Once the user is successfully registered he/she can login to the app. The user is then navigated to the respective home screen based on whether he/she is a landlord/tenant as shown in Figure 4.2.

Figure 4.2 Landlord Home Screen
4.2 Landlord

The Landlord Home Screen contains a list of issues being reported by the tenants. This list is shown using ListView and floating action button is used for sorting the list view based on the fields new/in progress/done as shown in Figure 4.3.

Figure 4.3 Landlord Home Screen (sorted list)
4.2.1 Update Apartment

The Apartment tab contains of a ListView which contains a list of apartments under the landlord. When user presses the Home button in the toolbar, it will navigate to an activity where the landlord can update the apartments. The list view can be sorted based on fields Occupied/Vacant using the floating action button as shown in Figure 4.4.

![Figure 4.4 Update Apartment & List view](image-url)
4.2.2 Navigation Drawer (landlord)

The drawer toggle in the tool bar is used to open the navigation drawer. As the user clicks on the drawer toggle, a navigation drawer appears on the screen. It can also be accessed with the right swipe. The navigation drawer consists of Home, This month rent, Package Notifications, Utilities, Emergency Info, Documents and Logout fields as shown in Figure 4.5.

![Figure 4.5 Navigation Drawer](image)
4.3 Tenant

The Tenant home Screen consists of a list view of issues submitted by the tenant as shown in Figure 4.6.

![Figure 4.6 Tenant Home Screen](image)

Figure 4.6 Tenant Home Screen
4.3.1 Navigation Drawer (tenant)

The tenant's home screen also consists of a navigation drawer. The Navigation drawer consists of Home, This month rent, Package Notifications, Utilities, Emergency Info, Documents, Payments, View Documents and Logout fields as shown in Figure 4.7.

![List view in Navigation Drawer](image-url)

*Figure 4.7 List view in Navigation Drawer*
4.3.2 Report an issue form

The Floating action button will navigate to reporting an issue form which is shown in Figure 4.8, where the tenant can report an issue.

Figure 4.8 Report an issue form
4.3.3 Rent Payments

The user can make rent payments by entering valid card details as shown in Figure 4.9.

![Figure 4.9 Payments](image-url)
4.4 Push Notification

Push Notification will be sent to the tenant on data update as shown in Figure 4.10.
4.5 Cloud Server

The developed android app is connected to the cloud database server using Parse API. This cloud service lets user run applications using parse code which uses REST API for request and response to the web server. Figure 4.11 shows the code snippet for connectivity to the parse cloud from an app.

```java
package com.mycompany.mypro;
import android.app.Application;
import com.parse.Parse;
import com.parse.ParseACL;
import com.parse.ParseCrashReporting;
import com.parse.ParseInstallation;

public class ParseApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();

        // Initialize Crash Reporting.
        ParseCrashReporting.enable(this);

        // Enable Local Datastore.
        Parse.enableLocalDatastore(this);
        Parse.initialize(this, "SLLSG0nOKn0KSAN2cL6d9FwgyhNiBDQXLVAAkQVc", "6tPSB3tGxg6lkWyBPzyN2AfCjKIN9fiwTtv2iZYy");
        ParseInstallation.getCurrentInstallation().saveInBackground();

        ParseACL postACL = new ParseACL();
        postACL.setPublicReadAccess(true);
        postACL.setPublicWriteAccess(true);
    }
}
```

Figure 4.11 Snippet to connect with parse database
4.6 Cloud Code

Cloud code runs on the cloud rather than on a device which helps to update the code to all mobile environments instantly. Figure 4.12 shows the code to send push notification to the tenant on rent update.

```javascript
Parse.Cloud.define("sendPushToUser", function(request, response) {
    var current = request.params.cu;
    var aptnam = request.params.apt;
    var message = request.params.message;

    // Send the push.
    // Find devices associated with the recipient user

    var pushQuery = new Parse.Query(Parse.Installation);
    pushQuery.equalTo('user', current);
    pushQuery.equalTo('Apartment_name', aptnam);

    // Send the push notification to results of the query
    Parse.Push.send({
        where: pushQuery,
        data: {
            alert: "Your Package has arrived."
        }
    }).then(function() {
        response.success("Push was sent successfully.")
    }, function(error) {
        response.error("Push failed to send with error: " + error.message);
    });
});
```

Figure 4.12 Snippet to send Push Notifications
5 EVALUATION AND TESTING

To evaluate the application, the app is tested on Nexus 7 tablet with Android Version 5.01 (Lollipop). The app supports Android version 4.04 to 5.1. Each module of the application is tested with positive and negative test cases.

5.1 Registration

a. **Negative Test Case:** Clicking on ‘Sign up’ button without entering values in any of the fields, and trying to register with the application will be displaying a toast message “Fields are empty” as shown in Figure 5.1(a).

b. **Positive Test Case:** All details as shown in Figure 5.1(b) should be entered, and by clicking on ‘Sign up’ button the user will be registered successfully.
Figure 5.1 User Sign up Page
5.2 Login

a. **Negative Test Case:** When the user enters wrong credentials and tries to login, then an alert dialog displaying “Invalid Login credentials” as shown in Figure 5.2(a).

b. **Positive Test Case:** By entering valid credentials as shown in Figure 5.2(b) and clicking on the ‘Sign in’ button the user is authenticated and navigated to the Home screen.

![Figure 5.2 User Login Page](image-url)
5.3 Forgot Password

a. **Negative Test Case:** When the user doesn’t enter a valid email id, a Toast message displaying “Please enter your mail id” as shown in Figure 5.3(a).

b. **Positive Test Case:** By entering valid email id and clicking on the ‘Reset Password’ button as shown in Figure 5.3(b) the parse server will send an email to the user’s account as shown in Figure 5.4(a). The user can click on the link which will navigate to a web page as shown in Figure 5.4(b). When the user clicks on the ‘Change password’ Button, the password will be reset and a success message is displayed as shown in Figure 5.4(c).

![Figure 5.3 Reset Password](image-url)
Figure 5.4 Password link to inbox
5.4 Floating Icon in Issues Page

As the landlord is logged into the application, the home page of the landlord displays the issues that are updated by the tenant as shown in Figure 5.5.

Figure 5.5 Floating Icon
The landlord can sort the issues by clicking on the floating icon button shown in Figure 5.5, where three other buttons pops out to sort the list based on ‘new’ as shown in Figure 5.6(a), ‘In Progress’ as shown in Figure 5.6(b) and ‘Done’ as shown in Figure 5.6(c).
5.5 Add Apartment

a. **Negative Test Case:** When the user doesn’t enter any of the fields, an alert message displaying “Fields Should not be empty” as shown in Figure 5.7(a).

b. **Positive Test Case:** By entering valid details and clicking on the ‘Update Apartment’ button as shown in Figure 5.7(b), the data will be stored on parse server.

![Figure 5.7 Add Apartment](image)
5.6 Apartments List

The home page also consist of apartments tab in a sliding tab layout which displays the list of apartments updated by the landlord as shown in the Figure 5.8.

Figure 5.8 List of Apartments
The landlord can sort the issues by clicking on the floating icon button where three other buttons pops out to sort the list based on ‘Vacant’, ‘Occupied’ and ascending order based on dates as shown in Figure 5.9 (a), (b), (c).

Figure 5.9 Sort List of Apartments
5.7 Apartment Details

When an apartment is added as shown in Figure 5.7(b), the list is updated as shown in Figure 5.10(a). When the list item is clicked the user is navigated to the apartment details screen as shown in Figure 5.10(b), where the landlord can delete the apartment. As the tenant vacates the apartment a new user id and password should be created. The landlord can also change the status of the apartment to ‘occupied’ or ‘vacant’, by clicking on ‘Apartment Status’ button. By clicking on ‘Package Info’ button, the package information list is displayed as shown in Figure 5.11(a).

(a)                                                                                  (b)

Figure 5.10 Apartment Details
5.8 Package List

Figure 5.11(a) shows a list of all the package notifications. From here the landlord can change the status of the package to ‘Package Taken’ or ‘Package NOT Taken’ by clicking on the list item as shown in Figure 5.11(a) which displays a context menu as shown in Figure 5.11(b). The status is updated and displayed to landlord as well as the tenant.

![Figure 5.11 Package List](image)

(a) (b)
5.9 Delete Issue

As the issue is added by a tenant, the list is updated as shown in Figure 5.12(a). On clicking the list item a dialog is displayed in Figure 5.12(b) confirming whether the user wants to delete the issue.

![Image of Rental Property Management app](a)

![Image of issue deletion dialog](b)

Figure 5.12 Issue Details
5.10 Update Status

The landlord can change the status of the issue to ‘In Progress’ or ‘Done’ as shown in Figure 5.13 and the same will be displayed to the tenant. This will help the tenant to keep track of the issues.
5.11 Zoom

The landlord can click on the image uploaded by the tenant to make it appear larger in pixels to get a clear view as shown in Figure 5.14.
5.12 List View

On clicking ‘This month rent’ in the navigation drawer as shown in Figure 5.15(a), a page with the list of the apartments is shown as in Figure 5.15(b).

(a)                                                                 (b)

Figure 5.15 This month Rent
5.13 Rent Update

a. **Negative Test Case:** When the user doesn’t enter any of the fields then an alert message displaying “Fields Should not be empty” is shown as Figure 5.16(a).

b. **Positive Test Case:** By entering valid details and clicking on the ‘Update Rent’ button as shown in Figure 5.16(b), the data will be stored on parse server.

![Figure 5.16 Update Rent](image)
5.14 Package Notification

On clicking ‘Package Notifications’ in navigation drawer as shown in Figure 5.15(a), a fragment is loaded as shown in Figure 5.15(b). On list item click for the respective apartment a package notification screen is displayed as shown in Figure 5.17(a). On clicking ‘Notify Package’ button an alert dialog displaying “Do you want to send Package Notification?” as shown in Figure 5.17(b). On clicking ‘ok’, a package notification will be sent to the tenant. On clicking ‘cancel’, the alert dialog disappears.

![Figure 5.17 Package Notifications](image)

(a) (b)

Figure 5.17 Package Notifications
5.15 Utilities Update

a. **Negative Test Case:** When the user doesn’t enter information for any of the fields and clicks on ‘Send’ button an alert dialog displaying “Fields Should not be empty” is shown as in Figure 5.18(a).

b. **Positive Test Case:** By entering valid details and clicking on the ‘Send’ button as shown in Figure 5.18(b), the data will be stored on parse server.

![Figure 5.18 Utilities bill](image)
5.16 Emergency info

a. **Negative Test Case:** When the user doesn’t enter information for any of the fields and clicks on ‘Submit’ button will display an alert dialog displaying “Fields Should not be empty” is shown as in Figure 5.19(a).

b. **Positive Test Case:** By entering valid details and clicking on the ‘Submit’ button the data will be stored on parse server as shown in Figure 5.19(b).

![Figure 5.19 Emergency Info](image-url)
5.17 Store Documents

On clicking documents in the navigation drawer as shown in the Figure 5.15(a), a fragment will be loaded as shown in the Figure 5.15(b). On list item click for the respective apartment a store documents screen is displayed as shown in Figure 5.20, where the landlord can upload the image files.

Figure 5.20 Documents
5.18 Tenant Login and Home Screen

When the tenant logs into the application as shown in Figure 5.21(a), the tenant will be presented with the tenant home page displaying the list of issues tenant has posted as shown in Figure 5.21(b).

![Tenant Login](image1)

![Home Screen](image2)

Figure 5.21 Tenant Home Screen (List of issues)
5.19 Escalate Issue

On clicking the issue in the list as shown in Figure 5.21(b) the user will be presented with the details about the issue as shown in Figure 5.22, where the user can escalate the issue or delete the issue.

Figure 5.22 Escalate Issue
5.20 Describe the Issue

As shown in Figure 5.23, the report issue screen consists of a form to describe the problem. The tenant can click ‘choose to click’ button to invoke camera and take a picture for uploading or click on ‘Upload from Gallery’ button to upload image from the gallery. The fields will be updated in parse database on clicking ‘Next Step’ button and ‘cancel’ will redirect the activity to home page.

![Figure 5.23 Describe Problem](image)

Figure 5.23 Describe Problem
5.20.1 Example

An example where the tenant is reporting an issue is shown in Figure 5.24(a) and 5.24(b).

(a)  
(b)  

Figure 5.24 Tenant reporting an issue
5.21 View Rent

On clicking ‘Rent’ in the navigation drawer as shown in Figure 5.15(a), the tenant gets a view of the rent posted by the landlord. The Sliding tab layout consists of ‘LATEST’ and ‘HISTORY’ tabs where latest contains the current rent posted as shown in Figure 5.25(a) and history consists of all the rents posted as shown in Figure 5.25(b).

![Figure 5.25 View Rent](image-url)
5.22 View Packages

On clicking ‘Package Notifications’ in the navigation drawer as shown in Figure 5.15(a), the tenant gets a view of the Package Notifications notified by the landlord. The Sliding tab layout consists of ‘PENDING’ and ‘HISTORY’ tabs where pending contains the packages that are not taken by the tenant as shown in Figure 5.26(a) and history consists of all the packages taken as shown in Figure 5.26(b).

Figure 5.26 View Packages
5.23 View Utilities

On clicking ‘Utilities’ in the navigation drawer as shown in Figure 5.15(a), the tenant gets a view of the Utilities posted by the landlord. The Sliding tab layout consists of ‘LATEST’ and ‘HISTORY’ tabs where latest contains the current Utilities bill posted as shown in Figure 5.27(a) and history consists of all the Utilities bill posted as shown in Figure 5.27(b).

![Figure 5.27 View Utilities](image)
5.24 Emergency Info

On clicking ‘Emergency info’ in the navigation drawer as shown in Figure 5.15(a), the tenant gets a view of the Emergency Information sent by the landlord as shown in Figure 5.28.

Figure 5.28 Emergency Info and Documents
5.25 View Documents

On clicking ‘Documents’ in the navigation drawer as shown in Figure 5.15(a), the tenant will be presented with the Sliding tab layout which consists of ‘View’ button to display files uploaded by landlord as shown in Figure 5.29.

![Figure 5.29 Documents](image_url)
5.26 Payments

a. **Negative Test Case:** When the user doesn’t enter any data in the fields, and clicks on ‘Make a Payment’ button, an alert message displaying “Fields Should not be empty” is shown as in Figure 5.30(a).

b. **Positive Test Case:** By entering valid details and clicking on ‘Make Payment’ button as shown in Figure 5.30(b), the data will be stored on parse server. Clicking on the ‘Pay full’ button will make full payment of the rent posted.

![Figure 5.30 Payments](a) ![Figure 5.30 Payments](b)
5.27 Push Notifications

As shown in the Figure 5.31, Push notifications will be sent to the tenant on Rent Update, Package Notification, Update Utilities Bill and on Sending Emergency information.

Figure 5.31 Push Notifications
6 CONCLUSION AND FUTURE WORK

Effectively resolving the apartment issues is important to the tenant's long-term future, the Rental Property Management app will be an important tool for creating rental housing stability by helping tenants speak with greater credibility through initiating and documenting communications and building productive relationships with landlords. Rental property management provides tenants of specific housing associations and social landlords with a simple way to report and arrange repairs to properties. Finally, the goal of the app is to create a better relationship between tenants and a landlord which can be achieved through this app.

Future work:

In the future this app:

- Can be extended to iOS Platform.
- Several apartments Database can be included.
- Could also allow local business to push deals/coupons within a certain geographic area.
- Adding persons to the apartment by selecting email id of the registered users.
REFERENCES


QR CODE

A machine-readable code consisting of an array of black and white squares, typically used for storing URLs or other information for reading by the camera on a smartphone [2].

(QR code for the URL of the Rental Property Management App Download)

App share link: http://penguin.tamucc.edu/~pshivraj/app-debug.apk