



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Mobile Computing

Android App
“Up and Away”

Project Whitepaper

Submitted By:

Monica Sivashankar (S0585851)

Sharanya Adiga (S0585838)

Pramod Shirale (S0586040)

Mohit Manoj Kokate (S0585841)

Application Description

“Travel and change of place impart new vigor to the mind” – Seneca. Human nature is such that it has a desire to see new places, it goes from one place to see another place which is called travel. Travelling also improves one’s understanding of other’s cultures, build social network and gives opportunity to disconnect from daily life. Some studies have shown that travelling improves one’s health.

After Covid-19 and continuous lockdowns, there is a boost in travel industry. In this paper, we are introducing a new app – ‘Up & Away’ - a travel guide application. This app provides an easy and convenient way for the user to view places, restaurants, and events in their city. The current scope of the application is within Berlin. This application uses technologies like Java, Android Studio and SQLite.

Features

- The application provides the user a guide to Berlin
- A user is able to view hotels, tourists’ places, restaurants, shopping centers, events
- Ratings, addresses and their respective websites can be viewed
- User can tap on the location and it launches Google Maps application so the user can navigate

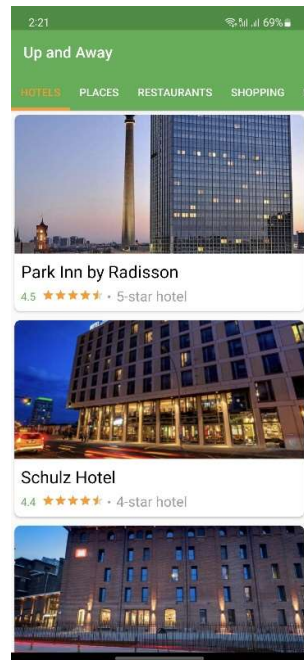
User Interface Design

1. Splash: This is the first screen the user can view when the app is launched



- Splash screen is one of the most vital screens in the application since it’s the user’s first experience with the application
- Splash screens are used to display some animations (typically of the application logo) and illustrations while some data for the next screens are fetched

2. Home Page: From the homepage, the user can see the tabs to switch



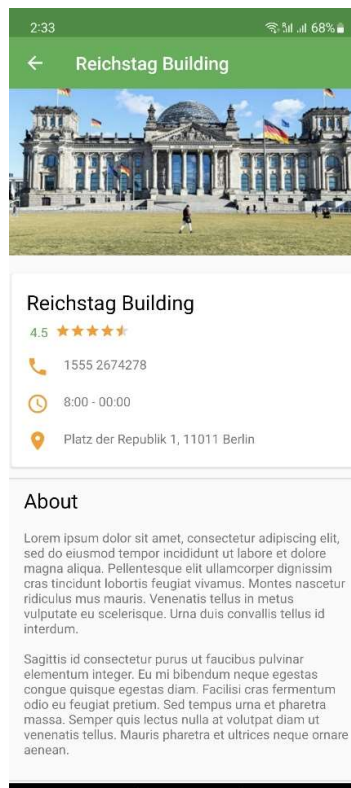
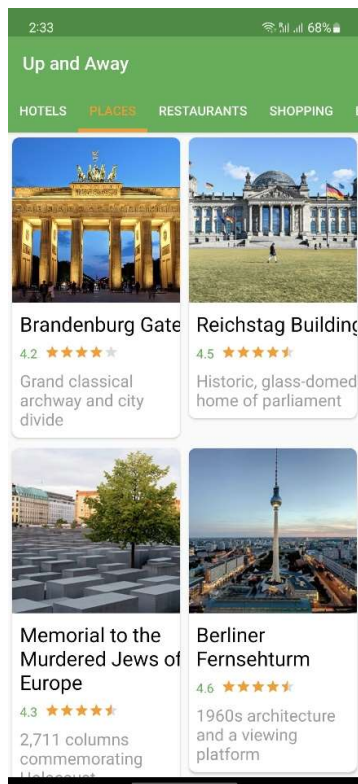
The user can see the following tabs to switch

- Hotels
- Places
- Restaurants
- Shopping
- Events

In this screen the user can see the various options for hotels in Berlin. The user can also view their ratings.

Upon, selecting one, the user can view the contact number, address and small description.

3. Places: From the homepage, the user can see the tabs to switch to places



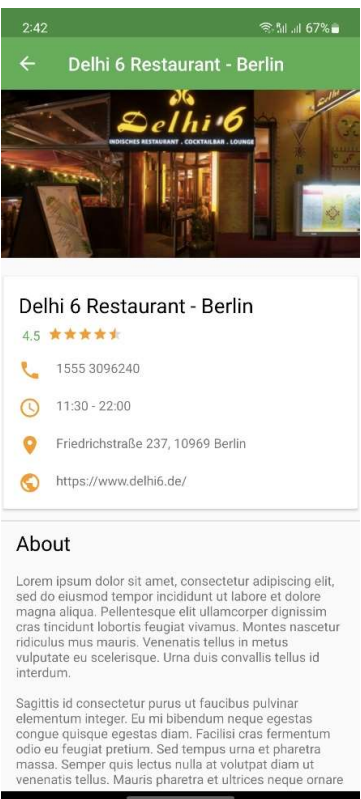
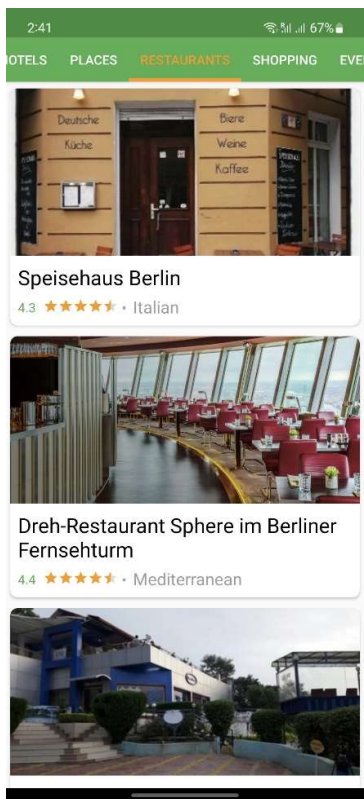
In this screen the user can see the various options for tourist places in Berlin.

The user can also view their ratings, small description of the place.

Selecting one the options can view the contact number, opening time and location.

The user can tap on the address shown and it launches Google Maps application so the user can navigate to the chosen location

4. Restaurants: From the homepage, the user switch to view popular restaurants in Berlin



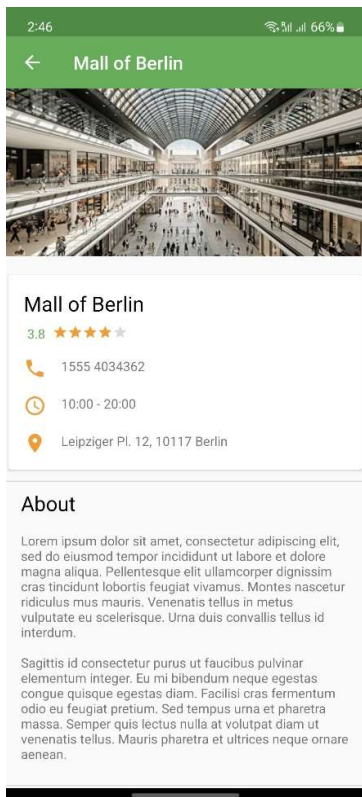
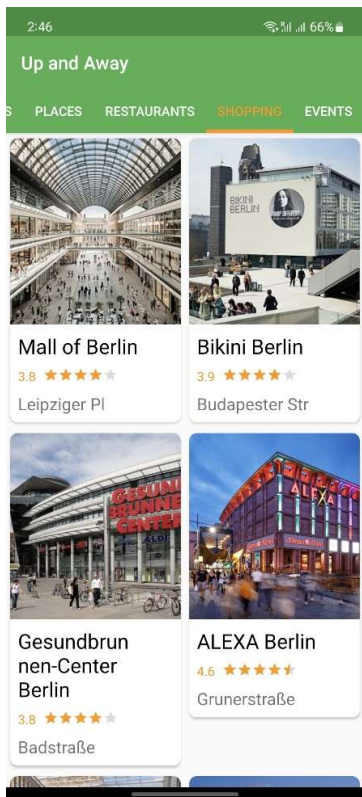
In this screen the user can see a few options for restaurants in Berlin.

The user can also view their ratings, small description of the place.

Upon, selecting one the options the user can view the contact number, opening time and location.

The user can tap on the website shown and it opens in the browser

5. Shopping: From the homepage, the user switch to view popular shopping centres in Berlin

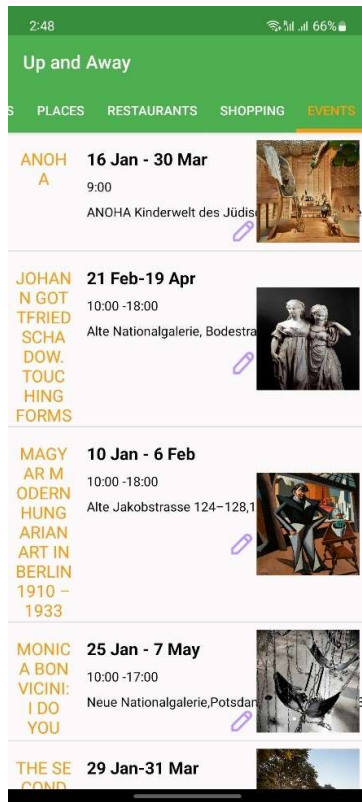


In this screen the user can see a few options for shopping in Berlin.

The user can also view their ratings and location.

Selecting one the options to view the contact number, opening time and address.

6. Events: From the homepage, the user switch to view popular events in Berlin



In this screen the user can see a few options for events in Berlin

The user can also view their ratings and location.

Selecting one of the events will navigate to the browser which shows the booking page of the respective event

The edit icon next to each event allows the user to change the timings in case of delay to update other users as well

Use Cases

1. A user can view different places of his interest like hotels, places, restaurants, shopping centers, and events. The user can select one of them and see the ratings, timings, location and description of the event selected.
2. The user can book tickets for an event by tapping on the one he/she selects. By clicking on the event, the browser is launched and navigated to the booking page.
3. In case of any delays or changes in timings, the user can edit and update the event time.

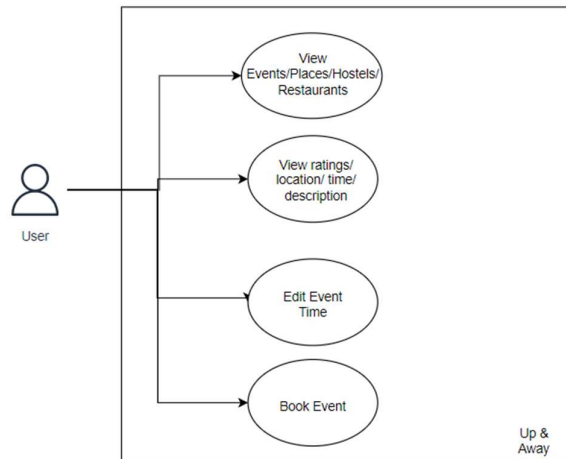


Figure 1: User Story

Software Architecture

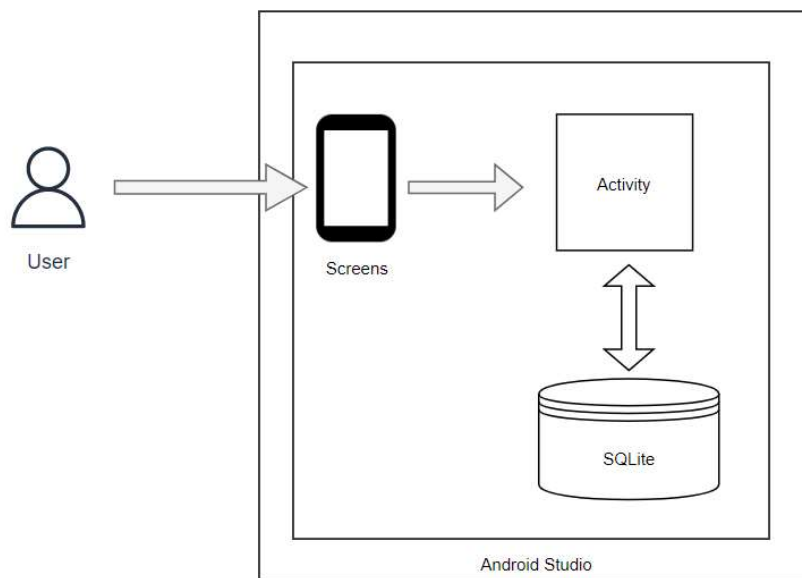


Figure 2: Software Architecture

Components

The main components of the application are:

- User Interface
- Database Layer

1. User Interface

This component is the layouts and screens used in the application such as homepage.

Inner Structure/ Algorithms:

The GUI design is written using XML Language in Android Studio. Each Layout is binded to a corresponding Activity which maps it with the functions to perform different app logics such as database calls.

AndroidViewModel

We have used AndroidViewModel to store the state of the UI and not to lose the data loaded in an activity while the user switches between the screens. ViewModel provides a convenient API for data persistence that resolves this issue.

NestedScrollView

We have used NestedScrollView widget to show the list of hotels, events, etc as a list that can be scrolled and viewed by the user.

CardView

We have used CardView widget to show the details of places and events as a card to be viewed by the user.

Screen	Test Case	Test Description	Status
Splash screen	View splash screen	Splash screen with logo should be seen	Passed
Home page	Switch between tabs	User should be able to switch back and forth between the different tabs	Passed
Hotels screen	Scroll and view hotels	User should be able to view list of hotels	Passed
Places screen	Scroll and view places	User should be able to view list of tourist places	Passed
Restaurants screen	Scroll and view restaurants	User should be able to view list of restaurants	Passed
Shopping screen	Scroll and view shopping centres	User should be able to view list of shopping centres	Passed
Events screen	Scroll and view events	User should be able to view list of events	Passed
Restaurants screen	View restaurant details	User should be able to view details and description	Passed
Restaurants screen	Open restaurant website	User should be able to view website on the browser	Passed
Event screen	Open event booking page	User should be able to view event's ticket booking page on the browser	Passed
Event screen	Open event location on Google Maps	User should be able to view event's location on Google Maps App	Passed

2. Database Layer

This component is responsible for storing and retrieving data. Since the Up and Away application stores large data, related to Berlin city. SQLite is known to be best for this use case as it is lightweight and natively supported in Android Studio. SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

The database is created and managed by a dedicated database helper class that extends the SQLiteOpenHelper class. This class provides methods for creating and upgrading the database, as well as inserting, updating, and deleting data. It also defines the database schema, including the tables, columns, and relationships between them.

The database component interacts with other components of the app, such as the user interface and the logic layer, to provide access to the stored data. For example, when the user edits the time of a particular event, the database component is responsible for inserting the new data into the appropriate table. Similarly, when the user wants to view any data from any of the tabs in the application, the database component retrieves the data from the appropriate table and provides it to the user interface.

The database component is an essential part of the application, as it allows the user to store and access the data. It provides a robust and reliable storage solution for the app, and makes it possible to add new features and functionality to the app in the future.

The database consists of 5 tables namely hotels, places, restaurants, shopping and events. All tables contain name, contact, address, website, ratings and time as columns and when user edits time it reflects only for events table. Schema for one of the tables is as shown below:

Hotel:

```
name TEXT
contact INT
address CHAR
website CHAR
ratings REAL
time REAL
```

Interfaces and Tests

EditTime Interface

Function to edit the time of a particular event by user and this interacts with the database.

Inner Structure/Algorithm:

Logic to get values from all the form fields of entry and store in the database.

```
public interface EditTime {
    interface View extends BaseView<Presenter> {
        void showAllEvent(); //show event list
        boolean isActive(); // returns true or false based on
        active status
    }
    interface Presenter extends BasePresenter{
        //save edited event entry:
        void editTime(E_name, E_time);
        void EditTimeToEventList(EventList e_list); // Adding edited
        event to list
        boolean isDataMissing(); //Boolean for null Data
    }
}
```

```

}
}

```

Test Case	Test Description	Status
Data validation	Correct data should be fetched for each tab from the DB	Passed
Edit Time	The time data of an event should be editable by the user	Passed
Updated Time	The edited time should be updated in the DB and change should be visible to the user	Passed

UML Sequence Diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

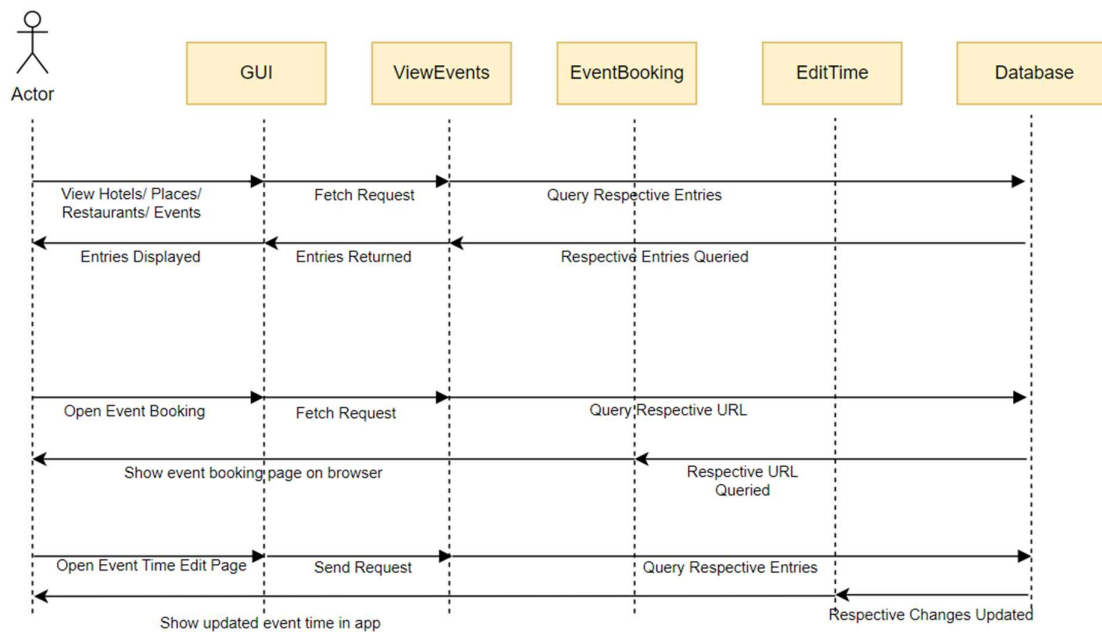
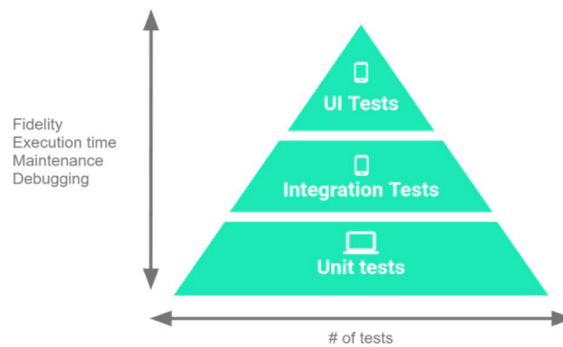


Figure 3: Sequence Diagram

The user can view hotels/places/ events, book an event or edit the time of event by sending a request to the database. The database uses the entries to query and the respective entries/ URL are sent back to the application and the user can view them.

End to End Testing



Testing Application:

We have planned to test our Up and Away app using Espresso Testing Framework. Espresso is an open-source android testing framework developed by Google. It is a simple, efficient and flexible testing framework.

The main components of Espresso include the following components:

- ViewMatchers: A collection of objects that implement the Matcher interface. It can be passed one or more of these to the onView() method to locate a view within the current view hierarchy.
- ViewActions: A collection of ViewAction objects that can be passed to the ViewInteraction.perform() method, such as click().
- ViewAssertions: A collection of ViewAssertion objects that can be passed the ViewInteraction.check() method. It uses a View matcher to assert the state of the currently selected view.

Features:

Small Core API, leading to ease in learning: It has a simple and lightweight API with three components: viewMatchers, viewActions and viewAssertions. Espresso's API makes Android UI tests easy to maintain and change.

Using Android support library:

```
android { defaultConfig {
testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}
}
dependencies {
testImplementation 'junit:junit:4.12'
androidTestImplementation 'com.android.support.test:runner:1.0.2'
androidTestImplementation 'com.android.support.test.espresso:espresso:3.0.2'
}
```

Using AndroidX library:

```
android { defaultConfig {
testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
```

```

}

}

dependencies {
testImplementation 'junit:junit:4.12'
androidTestImplementation 'com.androidx.test:runner:1.0.2' androidTestImplementation
'com.androidx.espresso:espresso-core:3.0.2'
}

```

Application's Manual Testing Results

Test Case	Test Description	Status
Scroll and View Events	User should be able to view all places, restaurants, etc and scroll through them in the app	Passed
Edit Time	The time data of an event should be editable by the user and view the updated time by the user	Passed
Book event on browser	User should be able to navigate to event booking page	Passed
Open event location on Google Maps	User should be able to view the location on Google Maps app	Passed

References

- 1) Android Studio Developer guides (<https://developer.android.com/docs>)
- 2) SQLiteDB documentation (<https://www.sqlite.org/>)