

NORTH SOUTH UNIVERSITY

Department of Electrical and Computer Engineering



CSE327.5

BashaBari

Submitted By:

Md. Nasir Uddin 1811274642

Md. Mahfuzur Rahman Shakil 1812918642

Submitted To:

AKM BAHALUL HAQUE

Lecturer

Department of Electrical and Computer Engineering

INDEX

Chapter

Cover Page	1
1. INTRODUCTION	4
1.1 Motivation	4
1.2 Proposed System	5
1.3 Market analysis	6
2. REQUIREMENTS ANALYSIS AND SYSTEM SPECIFICATIONS	7
2.1 System Requirement Analysis	7
2.2 Validation	10
3. SYSTEM DESIGN	11
3.1 Design Approach	11
3.2 Use Case Diagram.....	12
3.3 Sequence Diagram.....	13
3.4 Class Diagram	15
3.5 User Interface Design	16
3.6 Database Design.....	28
3.7 Entity Relationship Diagram	30
4. IMPLEMENTATION, TESTING AND MAINTENANCE	31
4.1 Introduction to Programming Languages, IDE's, Tools	31

4.2 Programming Languages.....	31
4.3 IDE'S, Tools and Technologies.....	32
4.4 Security and Permissions in Android	33
4.5 Testing Plan.....	34
4.6 Test Cases	36
4.7 Application Maintenance.....	37
5. RESULTS AND DISCUSSIONS	38
5.1 Surveys.....	38
5.2 User Interface Representation	39
5.3 Design Inception.....	40
6. FUTURE WORK	41
7. CONCLUSION	42
8. Bibliography	43

1. Introduction

Our Project's name is BashaBari. In this android project, we are going to connect homeowners and tenants with the help of a mobile app. In the past, communication between homeowners and tenants happened with emails, phone calls, and text messages. These sorts of communication are inefficient and overwhelming for those managing an outsized number of units. Everything is kept in one place in BashaBari application. No longer do homeowners and tenants need to send through emails and text messages to remember what was said. Not only is chat, it's also many more so that you can find what you're looking for with ease. Homeowners and tenants can communicate with each other, send notices, send requests, pay bills and accomplish some other tasks through this app.

1.1 Motivation

In urban areas, it can often be rather hard to maintain various things in an apartment. For example, often, it is a challenging task to identify particular problems and communicating with certain tenants can be awkward too. Homeowners often face difficulty in tracking the bills when there are dues left from previous months. one of the foremost challenging aspects of being a landlord or a tenant is that the communication breakdown between the two parties or playing email or phone to undertake and resolve a problem. This is often why BashaBari software won't only assist you in collecting information about tenants but also provide some channel of tenant communication. All these problems can be avoided by using our apps "BashaBari".

1.2 Proposed System

The Proposed solution is an Android Mobile Application, “BashaBari” which tries to avoid the above-mentioned limitations by standing communication between homeowners and tenants. BashaBari is an application that is designed and developed in such a way that it can run on any mobile device, smartphone, or tablet. It is an Android mobile application that runs on an Android platform with Internet support. In this project, the BashaBari Android application helps homeowner by:

- Add tenants with their details information like name, flat no, NID number, phone number.
- Only owner can add their tenants personally to their account.
- Send Notice to tenants
- Realtime chat with tenants
- Get any kinds of requests and reasons from tenants
- The owner can update their profile information quickly and also can change their password.
- Owners can easily log out from their accounts.

In this project, the BashaBari Android application helps tenants by:

- Tenants can receive registration through their homeowner.
- Tenants can update their profile information easily and also can change their passwords.
- Tenants can get a personal message from their owner.
- Tenants can get any kind of essential notices from their owner.
- Tenants can request any kinds of problems, reasons from their owner.
- Tenants can pay their bills, rent through our payment method.
- Tenants can easily log out from their accounts.

1.3 Market analysis

Target market

Our target market is Homeowners who will use “BashaBari” to manage their tenants more effectively and also accomplish certain tasks more easily with our app. “BashaBari” can be a very useful app for homeowners if it is used in the right way.

Competitor analysis

“BashaBari” is a unique concept and currently has no real competition in the market. But we are aware of the fact that if we do succeed with this app, we will have to face some competition pretty soon.

2. REQUIREMENTS ANALYSIS

2.1 System Requirement Analysis

The processes used for requirements engineering vary widely depending on the application domain, the people involved and the organization developing the requirements. However, there are a number of generic activities common to all processes

- i) Requirements elicitation
- ii) Requirements analysis
- iii) Requirements validation
- iv) Requirements management

We try to follow these steps when we create Bashabari application.

1) Functional Requirement: Functional requirements are properties that must exist within the final system. Functional requirement of bashabari application:

- i) Owner and tenant registration
- ii) Owner and Tenant Login
- iii) Manage owner and tenant information
- iv) Owner home page with owner information and request's from tenants.
Tenant home page with tenant information and notice's from owner.
- v) Navigation drawer layout Menu bar in both owner and tenants section from where they can navigate other pages.
- vi) Owner:
 - a) Send message and notice to tenants.
 - b) Register a new tenant.

- c) Send monthly bill to tenants.
- d) A complete tenant list with tenant information.
- e) Update information.
- e) owner logout.

Vii) Tenant:

- a) Send request to owner.
- b) New message from owner.
- c) Monthly bill information.
- d) Send bill via selected method.
- e) Update information.
- f) Upload NID image.
- g) Tenant logout.

viii) See the information about application in about us section.

ix) Share the application in share section.

2) Non Functional: Nonfunctional requirement specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. There are many types of nonfunctional requirement like Scalability, Reliability, Availability, Capacity, Security, Environment, Data Integrity. We ensure the usability and effectiveness of the entire software system. As we follow nonfunctional requirement, we get benefits like good user experience, the reliability, availability, performance and formulating security policy of the software system.

3) Hardware requirement:

- i) i3 Processor Based Computer or higher
- ii) Memory: 4 GB RAM
- iii) Hard Drive: 50 GB
- iv) Internet Connection
- v) Android

4) Software requirement:

- i) Windows 7 or higher
- ii) Android development kit
- iii) Firebase account
- iv) Android 4 or higher

5) Data Requirements: The set of data or information that's involved in project is defined using data requirements. For Bashabari project, the most data required is that the login information to register the application. Without registration, user will not get the complete feature of our application.

6) Performance Requirements: Response time, scalability, platform dependencies, tolerance are the performance requirements that ought to be considered when developing any system. The application or system should be able to respond quickly when the user interacts with the application. The application should be developed in such a way that it should be scalable enough to accept new features when we want to expand the application complexity. The application should run in all the specified software and hardware requirements from the design phase of the project. Also, the tolerance rate of the application should be at a higher level in

case of network issues and when the application crashes or stops. It should be able to deliver the information about any of those issues to the user when the system is no longer able to provide results when the user wants.

7) Testing and Maintainability Requirements: The application should be able to meet all the possible good and bad test cases under a test environment. Application should be developed in such a way that it does not have any issues or crashes when the user is using the application. It should be able to extend itself when we expand the code or implement any new functions to the existing application.

2.2 Validation

Validating any application is an important criterion before releasing the application to the users. If there is no validation, the information entered by users may be redundant, formatted inappropriately, and cannot be maintained. For example, we can validate the mobile number in a way that it should contain 11 digits. Suppose we don't check the validation; there are chances for the user to enter the wrong phone number and save it. In case the homeowner needs contact with the tenants, the owner cannot contact the respective person. Similarly, validations for all the fields that are used to save information in any application are highly necessary. In this application, we have done many validations in the Registration Page and Login Page.

In the Login Page, we have validated all the login information that is required for the user to sign up for the first time. Fields like username, password, mobile number are validated appropriately by displaying error messages. The username should not contain any digits, the password should be a minimum of six letters, the phone number should be a valid number, and if the phone number is already registered, an error message is shown saying that the phone number exists.

3. SYSTEM DESIGN

3.1. Design Approach

This project is based on the functional design approach, which helps understand the design of the project in a more straightforward way by explaining its flow, use cases, and implementation more like a modular approach. For example, there are different modules in this project which have separate functionality and other sub functionalities/modules. All the modules are designed, implemented, and integrated together to make a flawless working application.

3.2 Use Case Diagram

3.3 Sequence Diagram

Sequence Diagram of Registration

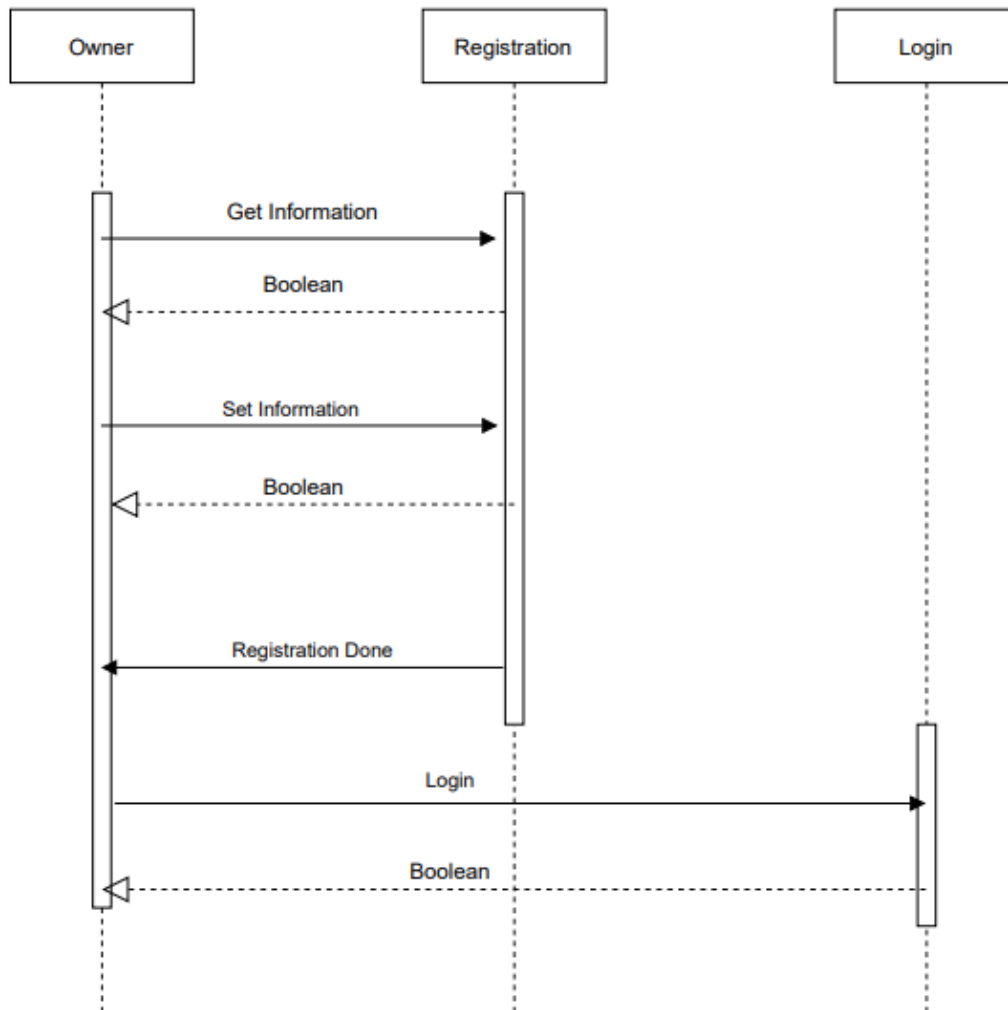
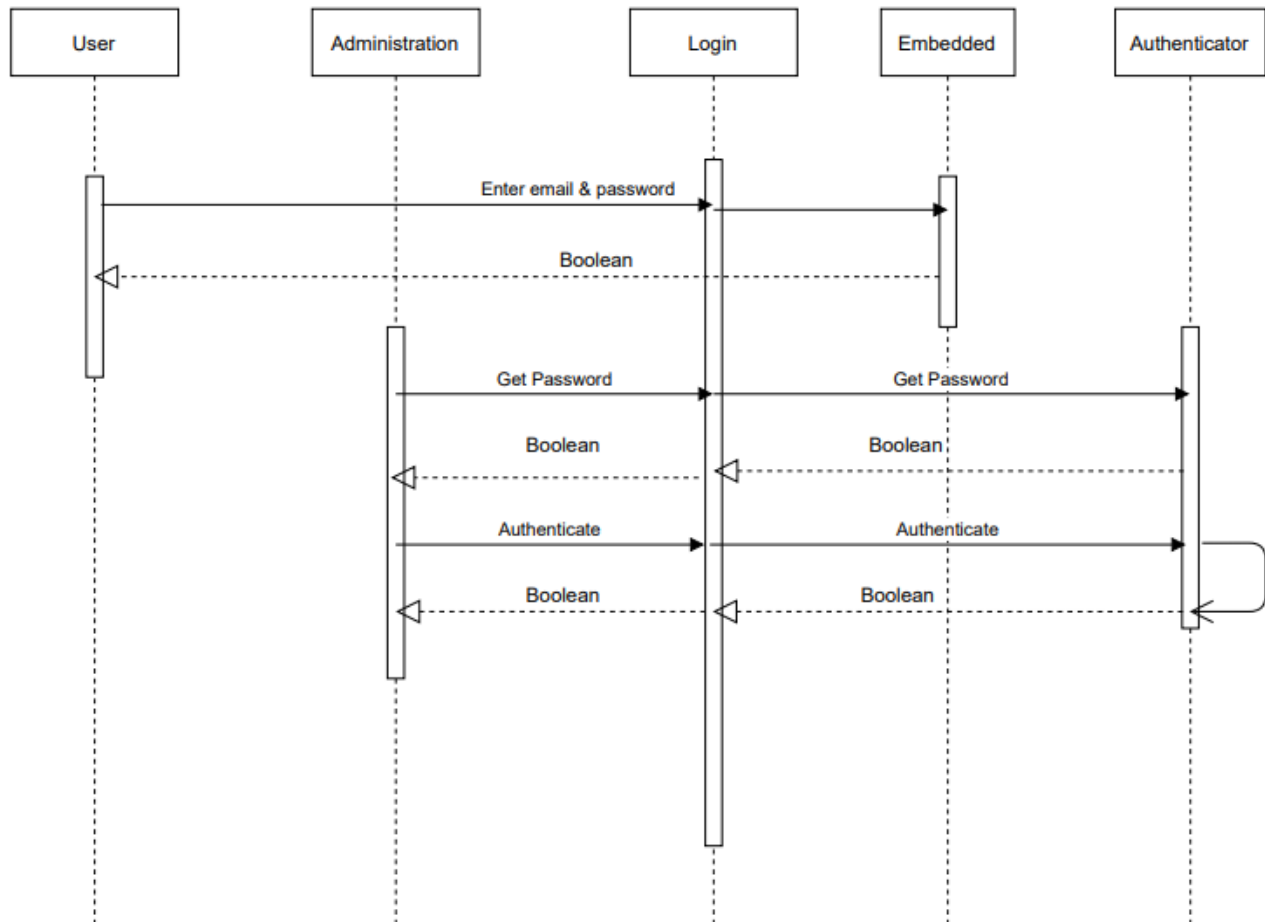


Figure2: Sequence Diagram for Registration

Sequence Diagram of Login

**Figure3:** Sequence Diagram for Login

BashaBari Sequence Diagram

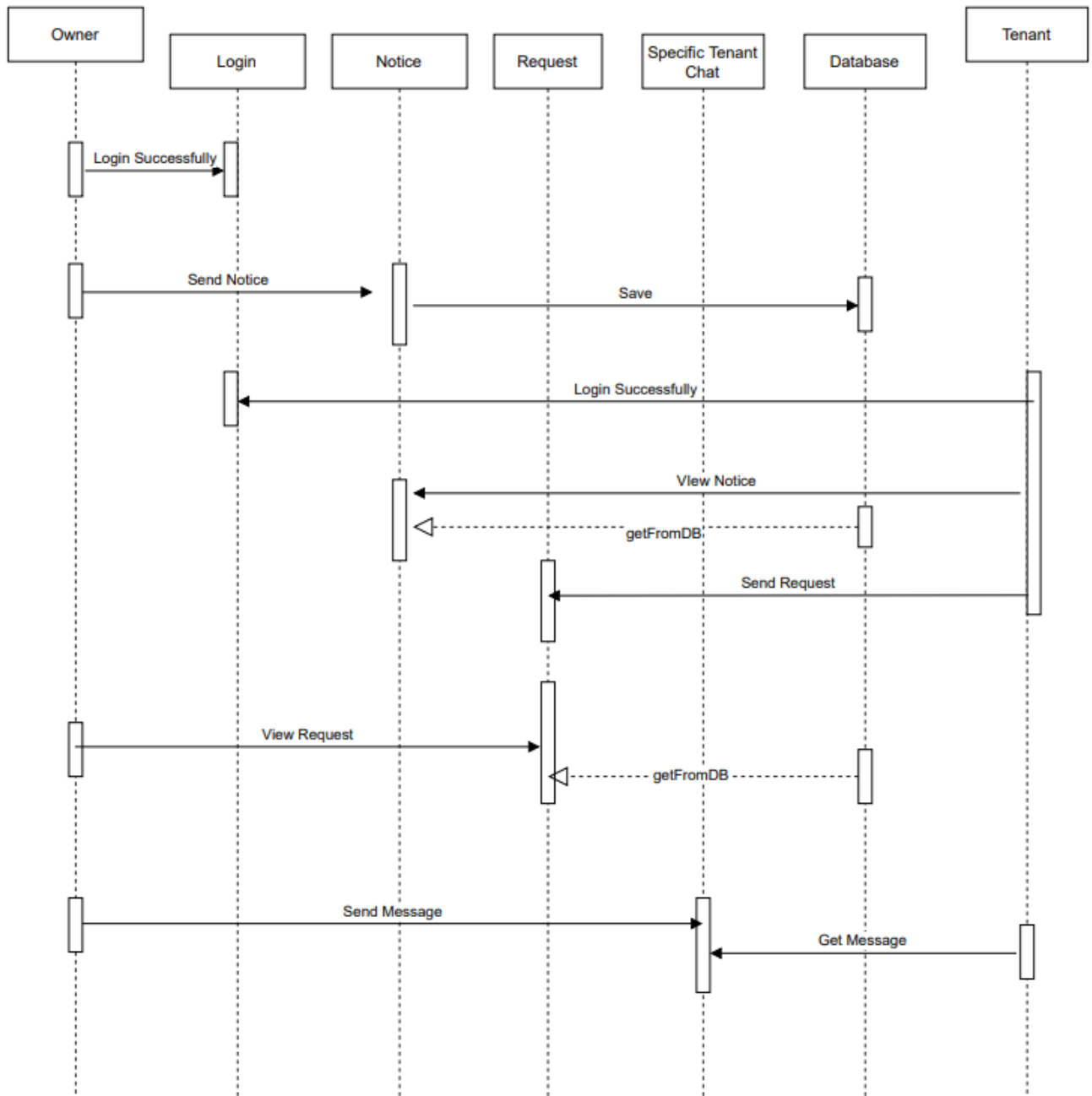
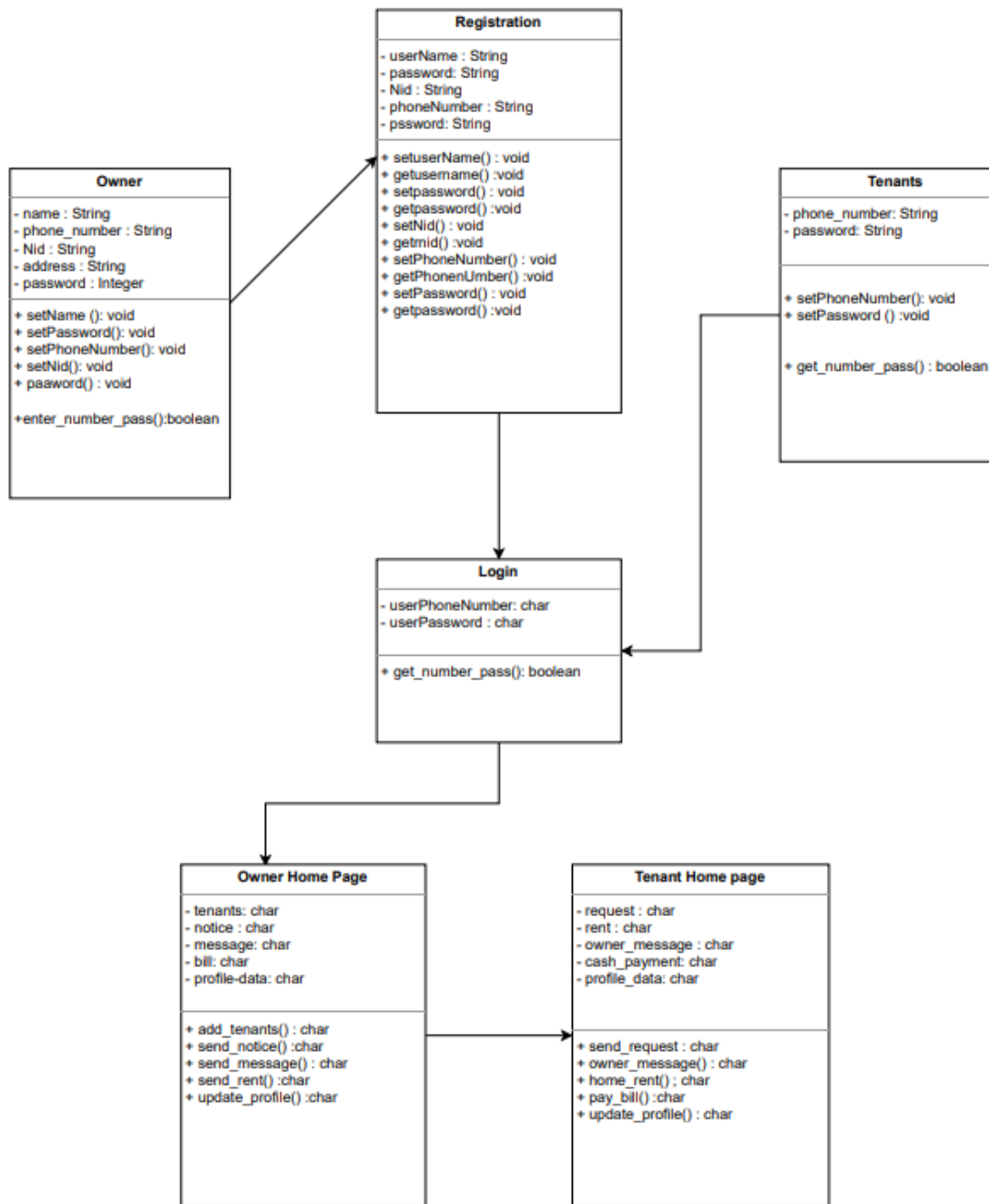


Figure4: Sequence Diagram

3.4 Class Diagram



Class Diagram
BashaBari

Figure5: Class Diagram

3.5 User Interface Design

Login & Registration Design

If the user wants to use the Bashabari application, they must install the application from play store. They have to register through register page. Registered information is stored on the firebase real-time database. After register, homeowner can login and can use our all feature of the application.

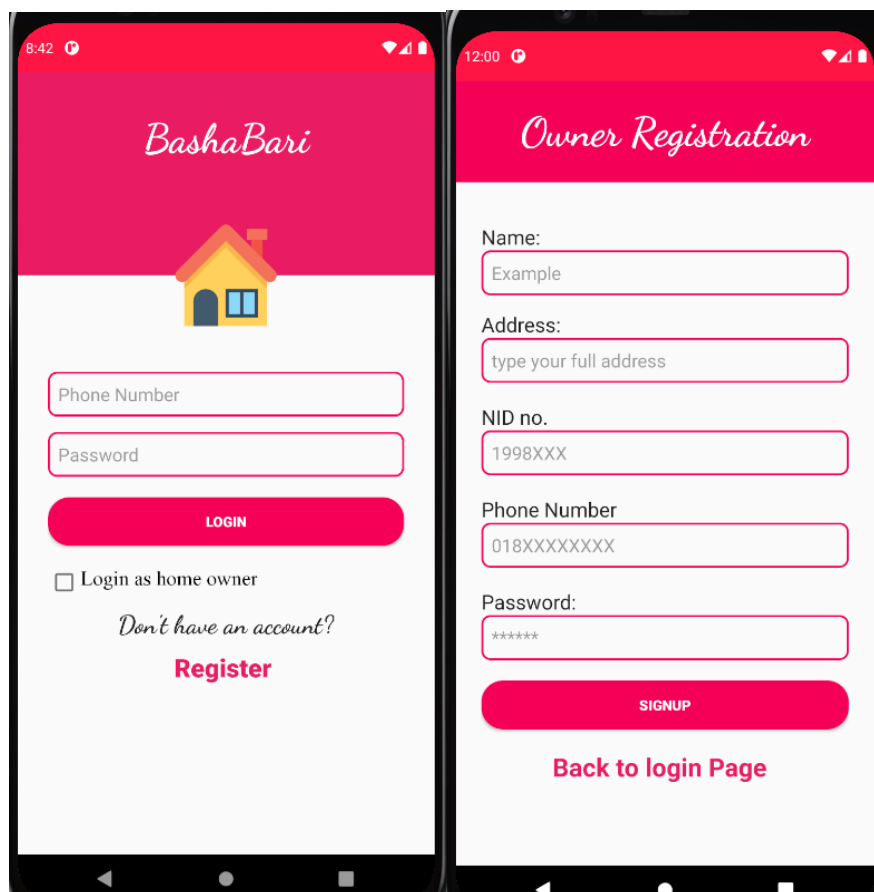


Figure6: Login & Registration

Owner System Design

1) Owner home page: There is a request section where recent request from tenants will appear. There is an also see more notice option where owner can see all request from tenants. In the home page, Owner name and home address also appear. To see request, we need to design and implement several java and xml file like Request java and xml, Request Adapter, Request Info and Request card view.

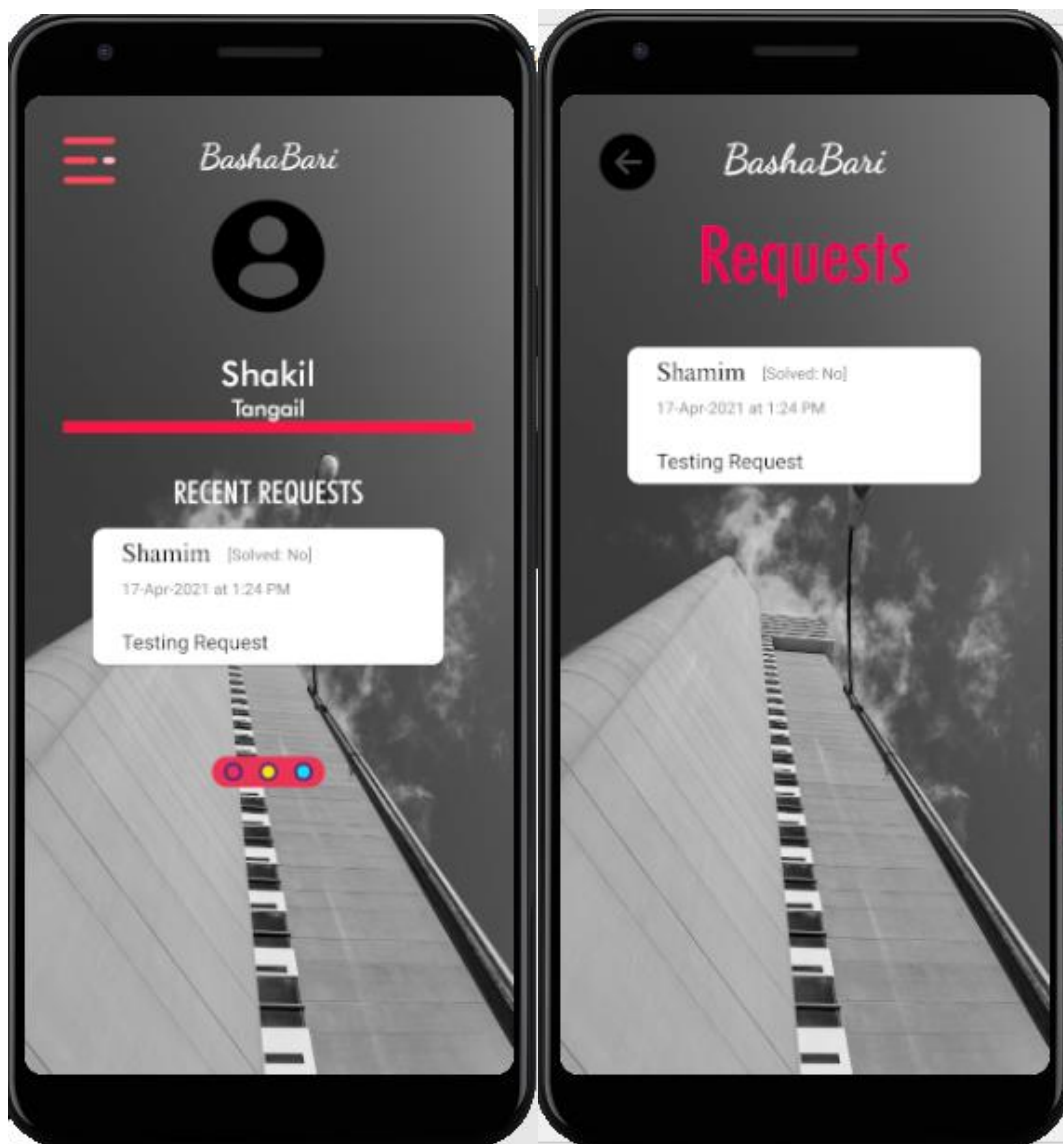


Figure7: Owner home page

2) Owner profile: In owner menu there is a Profile section. In profile section owner can see his details like name, phone number, national id number and password. This information we will get from Owner Database and owner also can update his information as well as he can change his password.

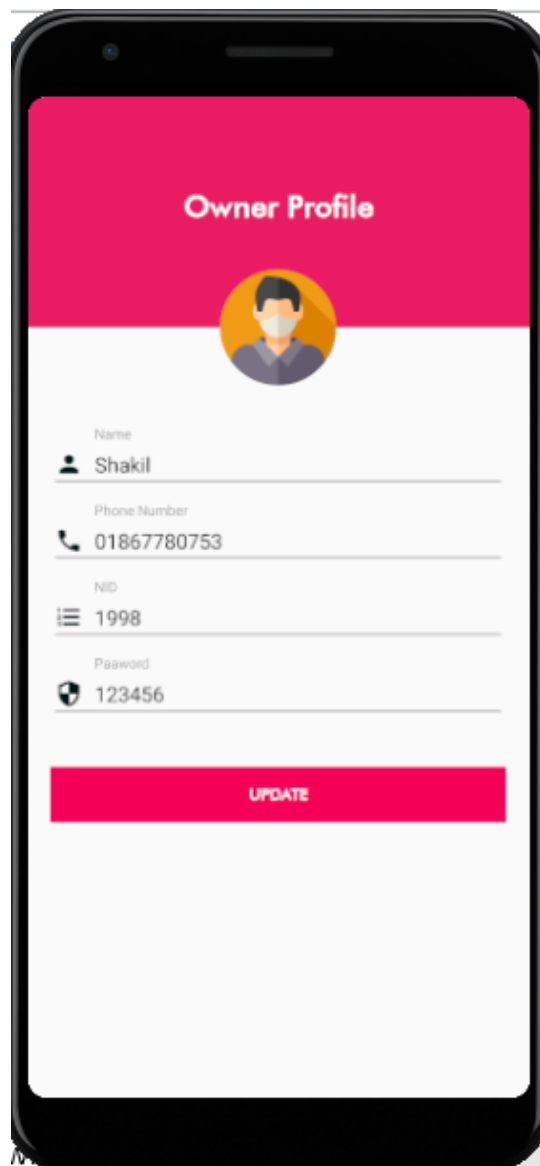


Figure8: Owner profile

3) Notice Section: In owner menu there is Notice section. In this section, Owner can send notice to tenants. And recent notice also appears in this section. Top of the layout we use edit text to write the notice and there is send option. After sending this information will store in Database. And Below send notice option, we use recycler view to show the recent notice of the owner.

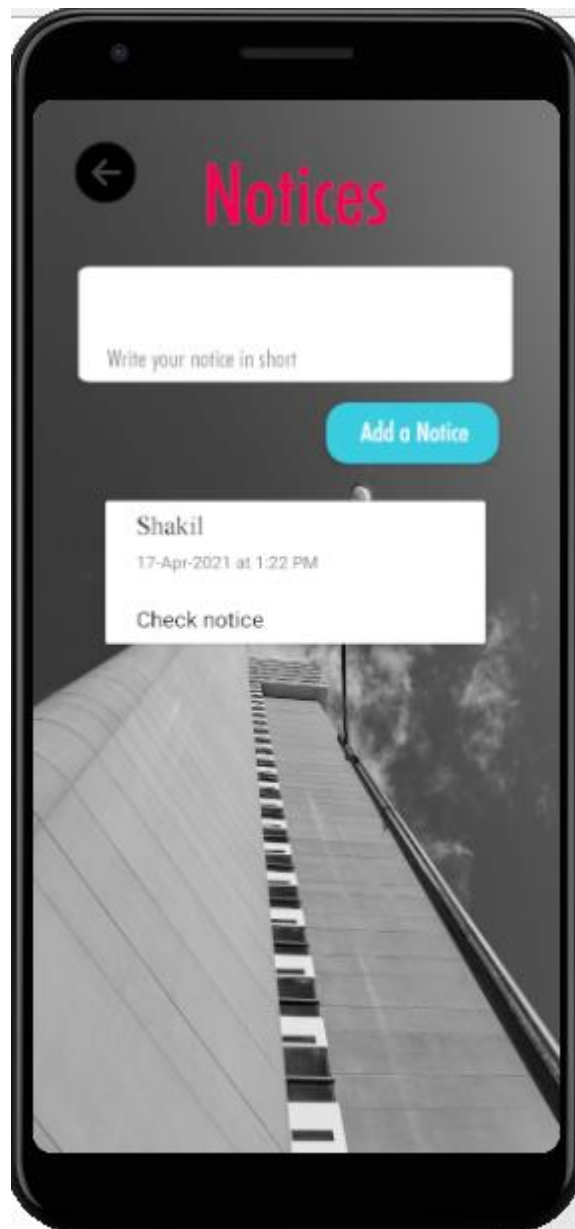


Figure9: Notice Section

4) Message section: There also a message section in owner menu. In this section, all tenants under owner will appear. If owner want to send any personal message rather than a notice. Owner can message any tenants. He can scroll the recycler list to find the tenants or there is search option in the top from where he can search for tenants. If he selects any tenant to send message, a new message section will appear from where he can send message.

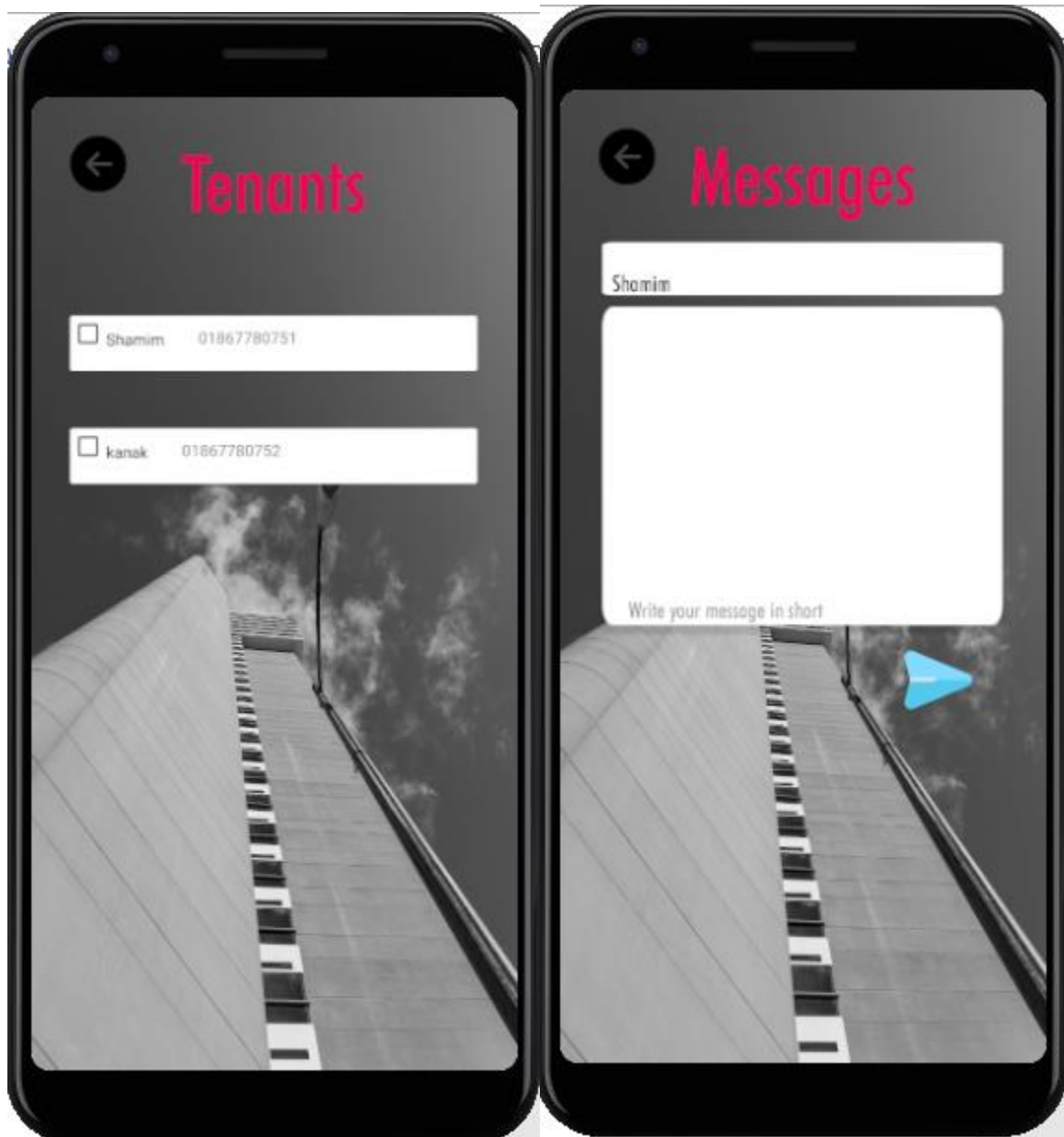
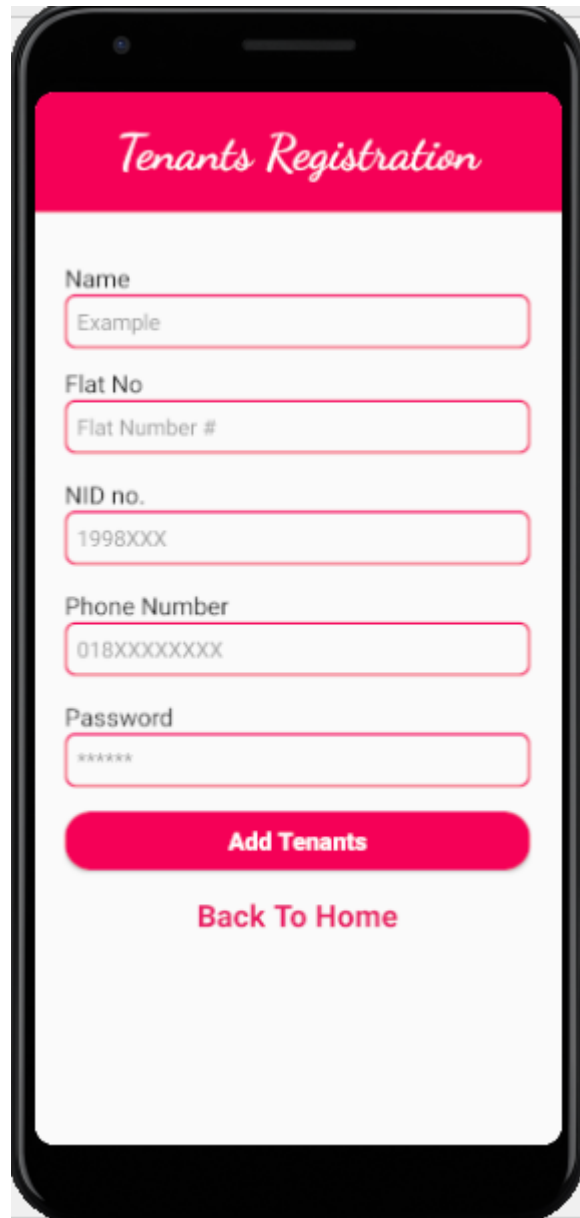


Figure10: Message section

5) Register Tenants: In the register tenant section, owner can add tenants. Owner has to provide tenants information like Name, Flat no, Nid no, Phone Number, password etc. This data will store in real-time database. After this registration, tenants can login in this system. For, tenants there is a different Home page, menu option etc.



Tenants Registration

Name
Example

Flat No
Flat Number #

NID no.
1998XXX

Phone Number
018XXXXXXXX

Password

Add Tenants

Back To Home

Figure11: Register Tenants

6) Bill section: There will a bill section in owner menu from where owner can send bill information to tenants monthly.

Tenants Design

1. Tenants Registration & Login: Tenants can receive registration through their homeowner. Homeowner collects their personal information like name, flat no, NID number, phone number and registers their tenants into our apps Bashabari. Then tenants can easily login into their accounts by using their phone number and password.

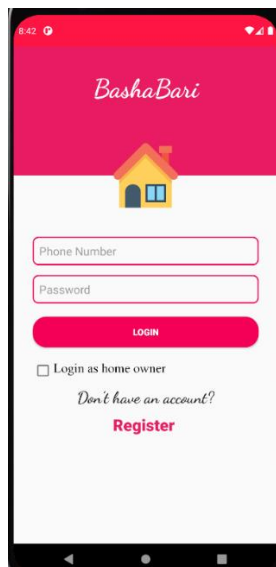


Figure12: Login Page

2. Update Profile: Tenants can update their profile information easily. They can also change their passwords which is given by their owner for security purposes. Here, we add some edit text for changing their necessary things.

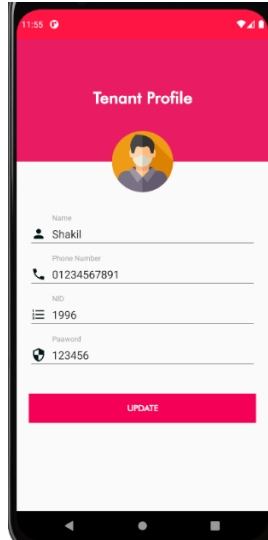


Figure13: Tenant Profile

3. Messages: Tenants can receive personal message from their owner. If owner want to communicate with one tenant for any kind of reasons, they can easily message them. This message will be show on the tenant's message section. Here we use recycler view to view messages which comes from database and show it here. We add a message card view where we design our message box.

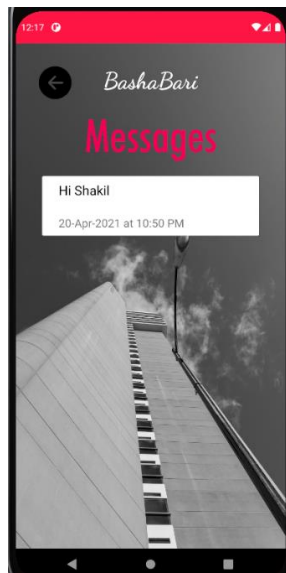


Figure14: Messages

4. Requests: Tenants can request any kinds of problems, reasons to their owner by through our requests section. Here, we design an edit text box where tenants can easily type their requests. We add a button 'add a request'. By pressing this button tenants can send their requests to owner easily.

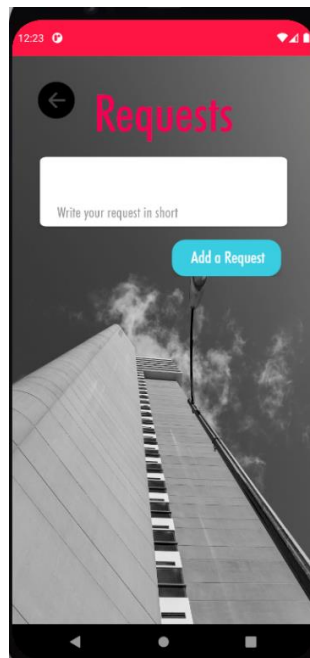


Figure15: Requests

5. Payment: We add a payment method by which tenants can pay their bills, rent easily. We add four kinds of payment methods. Those are Bkash, Nagad, Rocket, Visa, and MasterCard. Now, they can manually send money to their owner. The procedure of manually send money is adding in the individual pages of the payment method system.

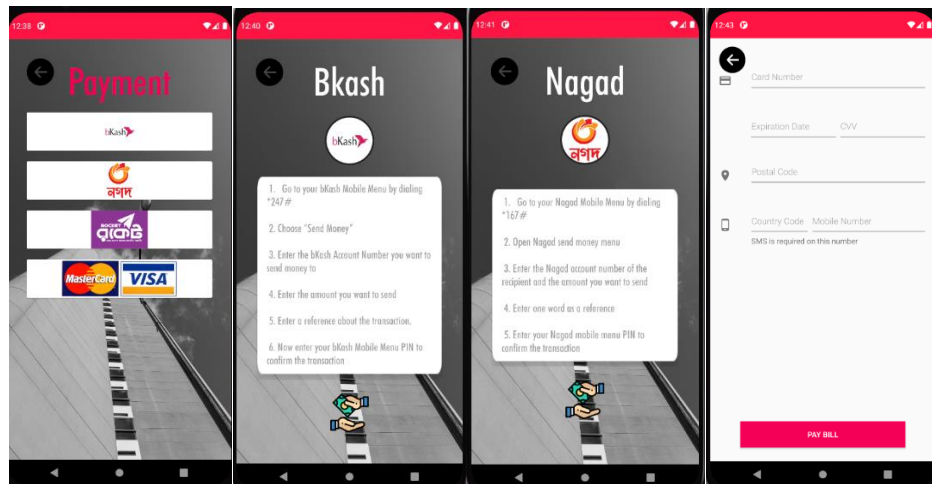


Figure16: Payment

6. Notice Section: There is a recent notice section where notice from the owner will appear. There is also see more notice button where tenants can see all notices from the owner. On the home page, the tenant's name and flat address also appear. Here we use recycler view to view notices which come from the database and show it here. To see notices, we have to design and implement java and XML files like notice java and XML, notice Adapter, notice Info, and notice card view.

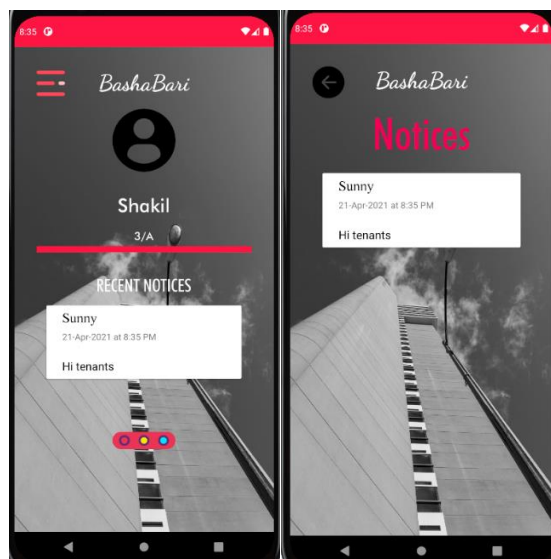


Figure17: Notices

7. Log out: Tenants can easily log out from their account by pressing Log out which is in the menu bar. Then it returns on the login page.

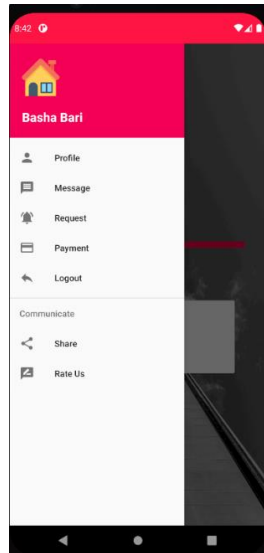


Figure18: Menu bar & Log Out

3.6 Database Design

The database should be designed in such a way that it should be easy to access and manipulate. Database definition and database manipulation operations should be performed accordingly to add, delete, and update values. In this project, we have used a Firebase database which is an open-source database, easy to find in Google. In Firebase, we use a realtime database. We add a JSON file in our android apps which we get from our Firebase to create the database. The Firebase Realtime Database is a cloud-hosted NoSQL database that lets us store and sync data between our users in realtime. The Firebase database server could be registered by providing our Gmail account. Here we use five tables for our databases. They are,

- Owner Database
- Tenants Database
- Notice Database
- Requests Database
- Message Database

The information to be transferred or fetched could be in JSON formats. In this project BashaBari, when we need any things to show like messages, notices, requests, the information is sent via internet services in JSON format in our recycler view format and displayed to users from Firebase real-time database.

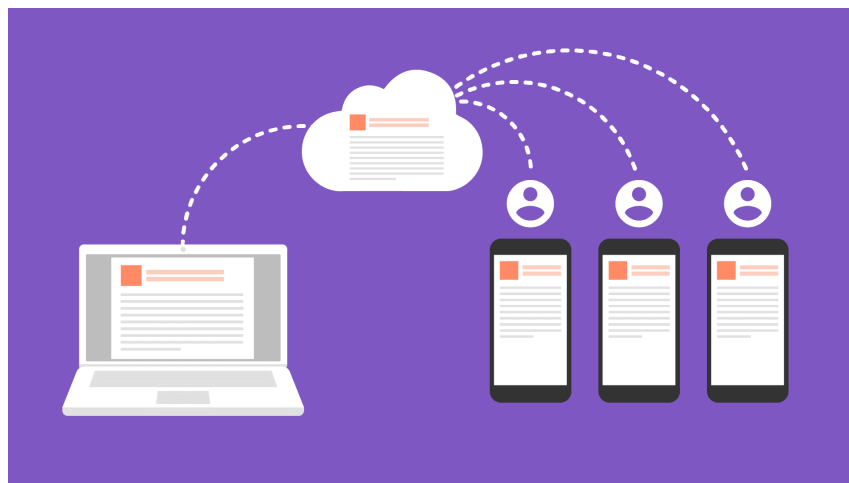


Figure19: Realtime Firebase Database

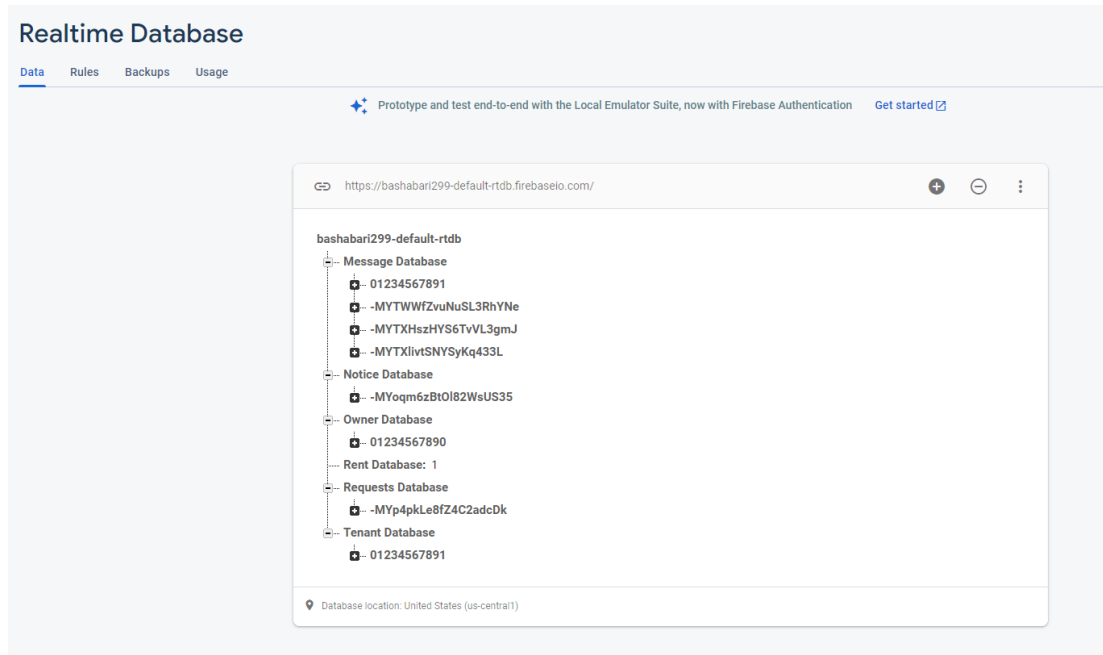


Figure20: Realtime Firebase Database of BashaBari

3.7 Entity Relationship Diagram

There are five tables involved for this project under one database (BashaBari), which was created in Firebase. As the five tables as per the design, the entity-relationship diagram is as follows:

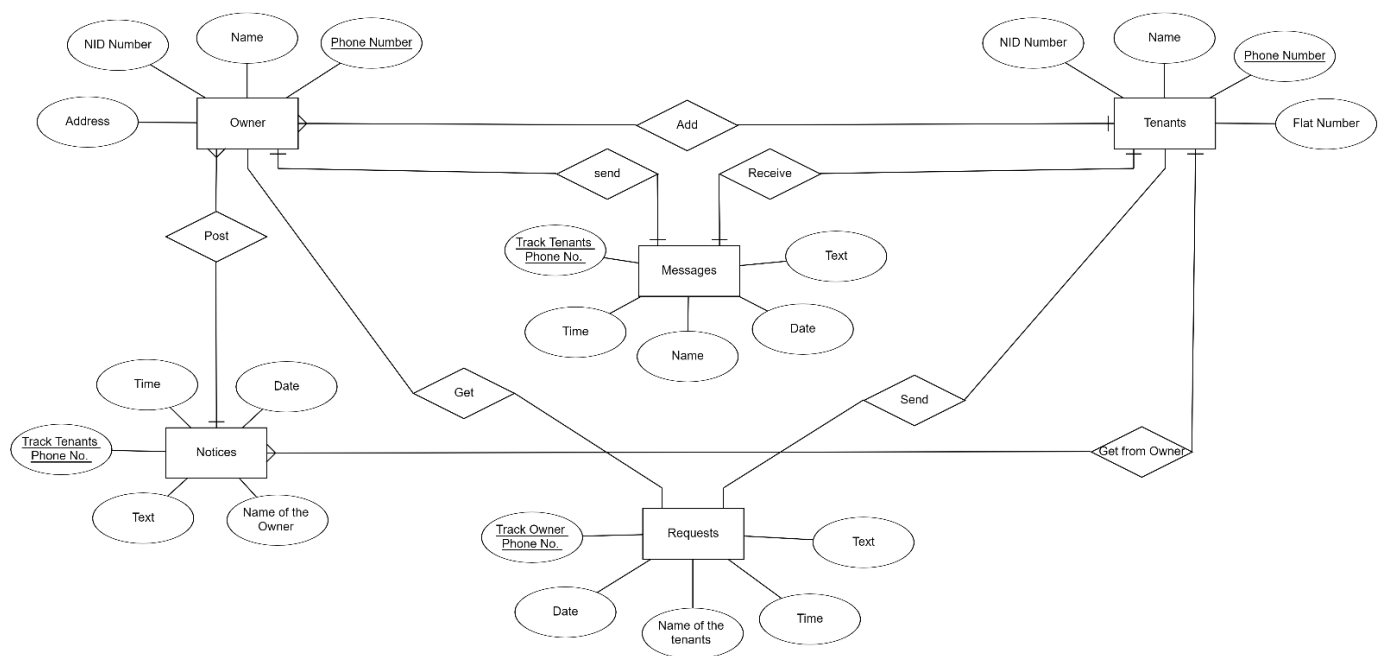


Figure21: ER Diagram

4. IMPLEMENTATION, TESTING AND MAINTENANCE

4.1 Introduction to Programming Languages, IDE'S, Tools and Technologies

The following programming Languages, IDE'S, Tools and Technologies used for this android project 'BashaBari' Implementation.

4.2 Programming Languages

Java: As the project is developing an Android Application, the default programming language is Java. All Android applications are built using Java in Android Studio. Java is a popular and widely used language throughout the world. As mentioned in, Java is one of the powerful programming languages like C, C++. developed by Sun Microsystems which has many powerful features as described below. After the development of C, C++, Java has come into evolution by addressing their drawbacks. It is one of the open-source projects that could be easily installed in our machine. The language is also easy to learn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. Many powerful features are supported by Java including various libraries, application services, graphics library for 2D/3D applications. The language is flexible enough to maintain code complexity, test, implementation, integration and support. Apart from these, there are other key features which make Java more special. It is object-oriented programming language, one of the important hierarchies in the programming languages which is used to implement real time applications, it provides for code reusability, it has a platform independence feature including any virtual machines, as in no need to write the many code for different OS as the Java Compilers convert the java source files to bytecode and this could be interpreted by any machine and the actual code is compiled irrespective of any machine, OS. It

is more secured as the compilers are designed efficiently to figure out any kind of errors.

4.3 IDE'S, Tools and Technologies

Android Studio: Android Studio is exclusively designed for developing Android applications. It consists of all Android SDK tools to design, develop, maintain, test, debug and publish our app. The IDE is designed very efficiently which makes the developer's job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, built-in functions or it could be anything else when we press the first letter of it. Say, suppose we declared few variables or methods that starts with an 'S', it automatically senses everything that starts with an 'S' and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are integrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any device and use the application. Other main tools include Android SDK, ADB, and Gradle Build.

Android Software Development Kit (SDK): One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

Gradle Build: Gradle Scripts are the recent feature that is added to Android Studio. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK's, version name, level, compiled SDK

version, build tools version. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

SDK Manager: It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or any other compatibility issues and to download any component that is required.

AVD Manager: It is used to create virtual devices of any desired API level to support higher level SDK's incase our device does not support. Using emulators to test the application is difficult as it might be little slower when compared to real device.

FireBase: We have used a Firebase database which is an open-source database, easy to find in Google. In Firebase, we use a realtime database. We add a JSON file in our android apps which we get from our Firebase to create the database. The Firebase Realtime Database is a cloud-hosted NoSQL database that lets us store and sync data between our users in realtime. The Firebase database server could be registered by providing our Gmail account.

4.4 Security and Permissions in Android

Security notions in Android are quite high. Whenever a new Android Application is created, a unique user and group ID. This makes the maintenance of the application in an easier way to avoid any security or privacy issues. As the application is created uniquely, it becomes private and no one can access other's applications. Permissions are another important concept which is included in

AndroidManifest.XML configuration file. This is required if the application wants to access the external features.

If the application wants to access the Internet, Camera or it could be any feature, it requires permissions. It is included within the tags as it is an XML file. Permissions are automatically created for the basic applications at the time when we create the application. If the app uses higher level API or SDK we must explicitly mention the permissions inside uses-permissions tag to access the features or components.

4.5 Testing Plan

Test plan is necessary for any project to plan the testing phase and decide the project's scope. Test plan involves collecting design specifications about the project, adding test cases, executing them manually or automatically using automated testing tools. Testing any application is highly important. The test plan is a method of documenting the test cases, specification plans, and other basic level details about how the application works.

1. Black Box testing: In this project BashaBari, sample test cases manual testing is done to check the functionality of the application and focus on ***what is the output?***.

2. White Box testing: Once the application meets the user requirements and functionalities according to the test cases, its internal logic are completely tested to ensure that the application does not have any logical errors or issues. Basically, here we focus on the internal mechanism ***how the output is achieved?***.

3. Unit Testing: We have tested all the modules of the application individually by running a test program.

4. Integration testing: After testing the modules individually, tested them by integrating all the submodules, modules into one application.

5. System Testing: It refers to checking whether the system in which the application is built meets the necessary requirements like software support. For example: In this project, I have checked whether the device in which the application developed

is compatible with the software (Android Studio) 29. Here, we also have security testing, recovery testing, performance testing.

6. End to End Testing: Tested the complete environment of the application by connecting the device with different machines, installing it as an APK file, with the database, and in the local network.

7. Usability Testing: Finally, usability testing is performed by testing the application's flow, UI design, and how flexible and easy the application is easy to use.

4.6 Test cases

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to the Apps	Apps should open	As Expected	Pass
2	Enter email & Password for Login	The credential can be entered	As Expected	Pass
3	Click Login	The user is logged in	As Expected	Pass
4	Select menu bar and press any button	All working	As Expected	Pass
5	Click see more notice/request button	Go to the more notice/request page	As Expected	Pass
6	Select payment method and confirm	Go to the payment methods	As Expected	Pass
7	Click pay bill	Go to the payment page	As Expected	Pass

4.7 Application Maintenance

Apart from designing and developing the application, maintaining the application is one of the important characteristics. The developer/owner of the application should be concerned about the maintenance of the application by fixing the issues. The solution for fixing such type of issues when the application crashes by using any of the phone number that are available for mobile applications.

5. RESULTS AND DISCUSSIONS

5.1 Surveys

This is very important to include our target users in every aspect. Because they are the reason why we are building our project application. With the data collected from a survey, we will be able to accelerate our process to a whole update level due to the insights and feedback given. We create some survey questions for the users.

- How important are the following aspects of our apps to you?
- How did you find our store?
- How do you like the apps design?
- How does the app 'BashaBari' run after the update?
- Is our app 'BashaBari' helping you achieve your goals?
- Are there any functions would you like us to add?
- Which of the following features do you use regular?
- What other features would you like to see us offer?
- How likely is it that you will recommend our apps to a friend or colleague?
- What is the reason for your score?
- What can we do to improve?

One of the important discussions following these results of the surveys was whether there was a great need for a messaging service within the app. Therefore, we decided that the app BashaBari would need to remove the need for a messaging service as much as possible. We would evaluate if a messaging service or chatbot should be added in the next iteration.

5.2 User Interface Representation

To make the application interactive, different controls have been used and designed using the layout file. Following are the important controls that are designed and used in this application:

Text View: The text view component belongs to the view group as a part of GUI.

It displays the text or content view of any activity to the user and allows them to edit.

- **Edit Text:** This allows itself to be editable in the text box.
- **Button:** One of the important components in which the application needs. It is mainly associated with action when the user clicks it. We can represent the button using any text and symbol which holds the action class on it.
- **Image Button:** Suppose, if we want to have an image for the button which we have designed, we can include using this control by adding the source or path of the image file within the tags in the layout file. All image buttons images are store in the drawable file.
- **Recycler View:** This is a key component under the view group which helps in displaying the information about anything when we click the action button. It also allows us to scroll through the screen and have a look at the information showed. Using this Recycler view adapter, the content is pulled from the database.
- **Checkbox:** It is the control component that allows us to use or make use of the function by just clicking on the check box button. When we include the check box widget in the application, we can see a small box on the screen on which we can check it, and it will be selected. We use it in our login and tenants list page.

5.3 Design Inception

We had begun to brainstorm about how the app should look and feel. 'BashBari' helps friendly and straightforward communication between landlords and tenants. We explored the potential moods and appearance individually through a design inception worksheet. We brought these ideas together, which culminated with:

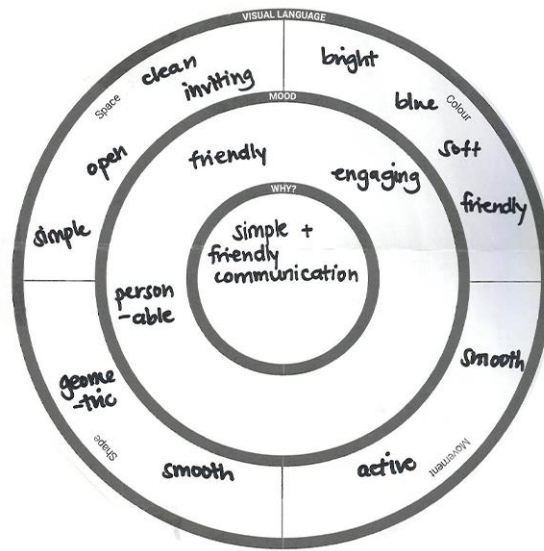


Figure22: Design Inception

Mood: Friendly, Engaging, Personable

Space: Simple, Open, Clean, Inviting

Shape: Geometric, Smooth

Colour: Bright, Blue, Soft, Friendly

Movement: Active, Smooth

6. FUTURE WORK

One of the main discussions following these results was whether there was a need for a messaging service with apps admin within the app. From the interviews, it was clear that constant communication from tenants aggravated landlords across the board. Therefore, we decided that the app would need to remove the need for a messaging service with admin as much as possible. We would evaluate if a messaging service or chatbot should be added in the next iteration. We also want to add a special feature that can help people to search for a home easily. Owner, who wants to rent his flat, they can post through our apps. The person who searches for an apartment, he can see all information about the apartment, and if he thinks the apartment is perfect for him, he can contact with the apartment owner quickly.

7. CONCLUSION

We have learned a lot from this project on how to develop Android Application and publishing it in real time by using Firebase. With an understanding of the user flow that was being developed by us, and with knowledge of the key functions that would be needed, we decided to get ahead by dividing up the key elements and starting to design some layout options.

Everything is becoming digital. So, why shouldn't we manage our tenants digitally? By using 'BashaBari' we can really change the way home owners manage their tenants. In conclusion, 'BashaBari' could be a useful addition to our daily applist and we will try to accomplish that to our best. Our BashaBari apps helps simple and friendly communication between landlords and tenants.

8. Bibliography

- "Agile Methodology," 4 July 2013. [Online]. Available: <https://www.codeproject.com/Articles/616070/Agile-Methodology>.
- P. Jain, "What Makes Java a Powerful Programming Language," 11 February 2013. [Online]. Available: <https://www.weblinkindia.net/blog/what-makes-java-a-powerful-programming-language>.
- A. Rongala, "Benefits of Java over Other Programming Languages," 7 May 2015. [Online]. Available: <https://www.invensis.net/blog/it/benefits-of-java-over-other-programming-languages/>.
- "Java SE Downloads," [Online]. Available: <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>.
- "Android Developers," [Online]. Available: <https://developer.android.com/index.html>.