Software Development Specification

For

EasyMed

(A smart pharmacy app)

Prepared by
Muhammad Abeer Tahmeed (1510963042)
Sohanur Rahman (1510464642)
Mehedi Rahman (1620647042)
Abdullah Al Nauman (1510041042)

North South University, Dhaka
Software Engineering (SE).

22, July 2019

Table of Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of this report is to give detailed description of the application for online medicine purchasing and ordering named EasyMed. It will clarify the main motive and functionalities of the proposed application, the interfaces of the system, what this application will do, etc. This document will also show the constrains of the app and how to overcome those obliges.

1.2 Intended Audience

This project is about online medicine ordering and purchasing. The intended audience is the honorable faculty member of Software Engineering course and the supervisor of this project Md. Musfique Anwar and the project managers & developers.

1.3 Intended Use

This project is intended to make buying medicine easy for everyone. By using this app anyone can buy medicine and deliver it to them without having to go to a pharmacy. Furthermore, anyone can see the description and side effects without confronting anyone.

1.4 Product Scope

The primary purpose of EasyMed is to purchase medicine by just a single click and to reduce the time consumption. This system will allow user to purchase the medicines and healthcare products through an app and get delivery at home 24/7. The user can also get symptoms-based suggestion and expert advice. This system will save a lot of time of the user. This app-based system will also serve patients in case of emergency.

Chapter 2

System Architecture Description

2.1 Overview of modules/components

- **IDE:** Android Studio
- **Language:**
  - **Frontend:** XML
  - **Backend:** Java
- **Code documentation tool:** Javadoc
- **API:**
  - Firebase API
  - Google Map SDK
  - Google Places API

- **Running platform:**
  - Android OS 4.0.3 Or better
  - 0.5 GB of RAM (1GB or more recommended)
  - Location settings must be available
  - A screen resolution of 800x480
- **Testing Framework:** Junit4

2.2 Database description

In our project, we will use Firebase real-time database for backend database. It is basically a NoSql database.

In short Firebase is platform which allow to build web and mobile applications without server-side programming language. Developers can store users data on its real-time database which sync data among users data in no time.

2.3 Class Libraries

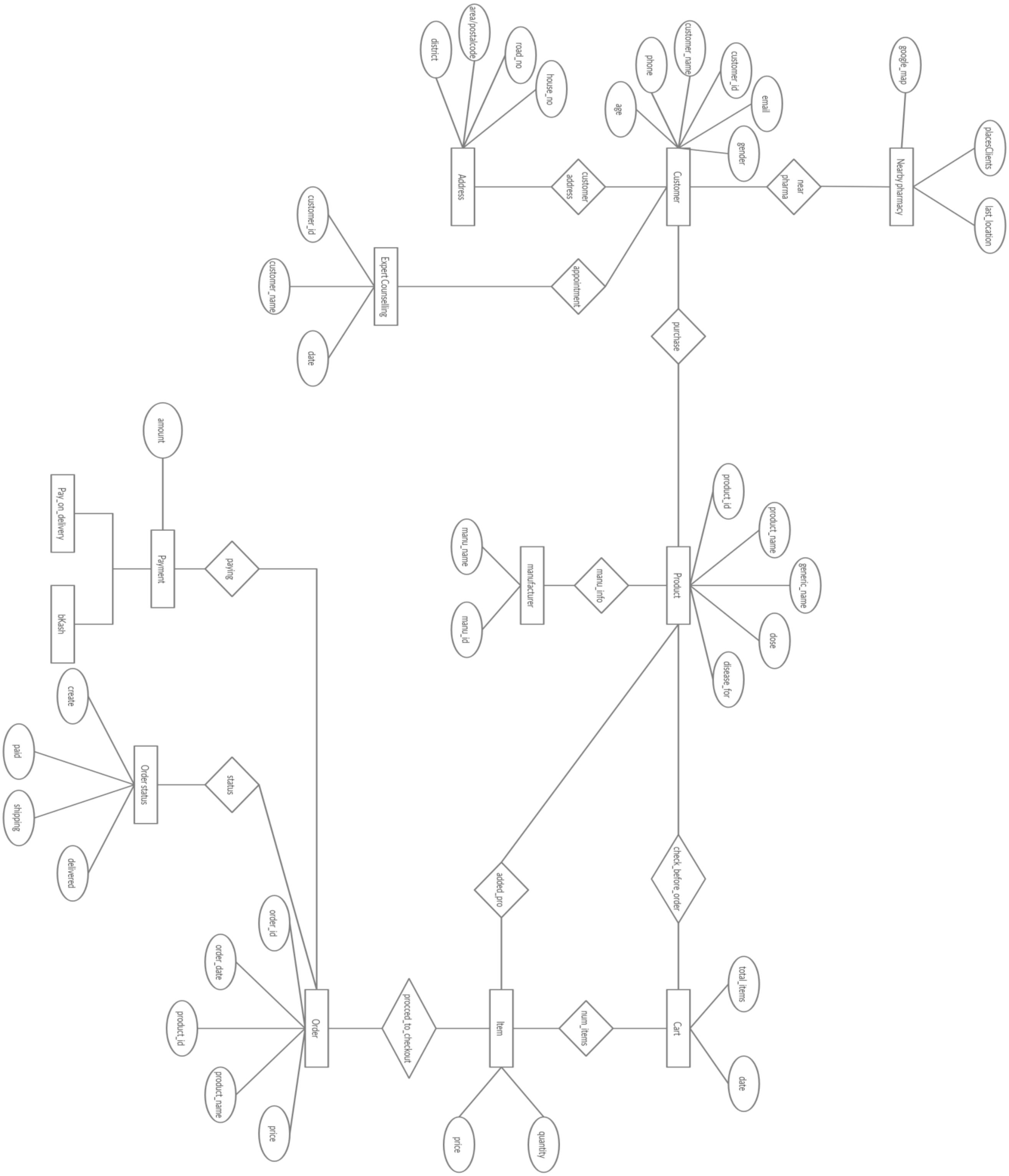
Dependency libraries:

1. **implementation 'com.google.firebase:firebase-auth:16.2.1'**
2. **implementation 'com.google.firebase:firebase-storage:16.1.0'**
3. **implementation 'com.google.firebase:firebase-database:16.1.0'**
4. **implementation 'com.google.firebase:firebase-core:16.0.8'**
5. **implementation 'com.firebaseui:firebase-ui-database:4.3.2'**
6. **implementation 'com.google.android.gms:play-services-maps:16.0.0'**
7. **implementation 'com.google.android.gms:play-services-location:16.0.0'**
8. **implementation 'com.github.mancj:MaterialSearchBar:0.8.2'**

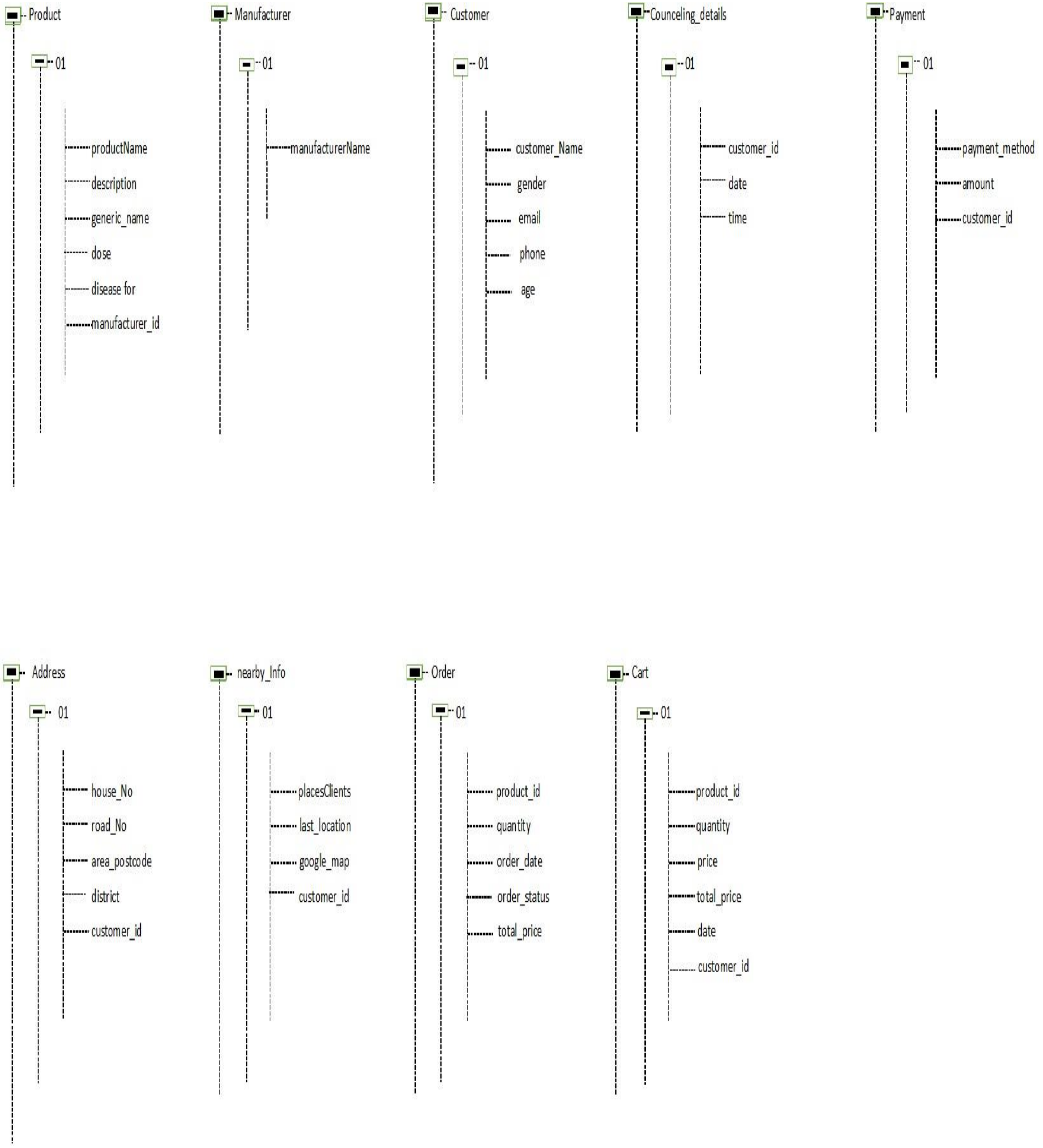
Chapter 3

Diagrams

3.1 E-R diagrams

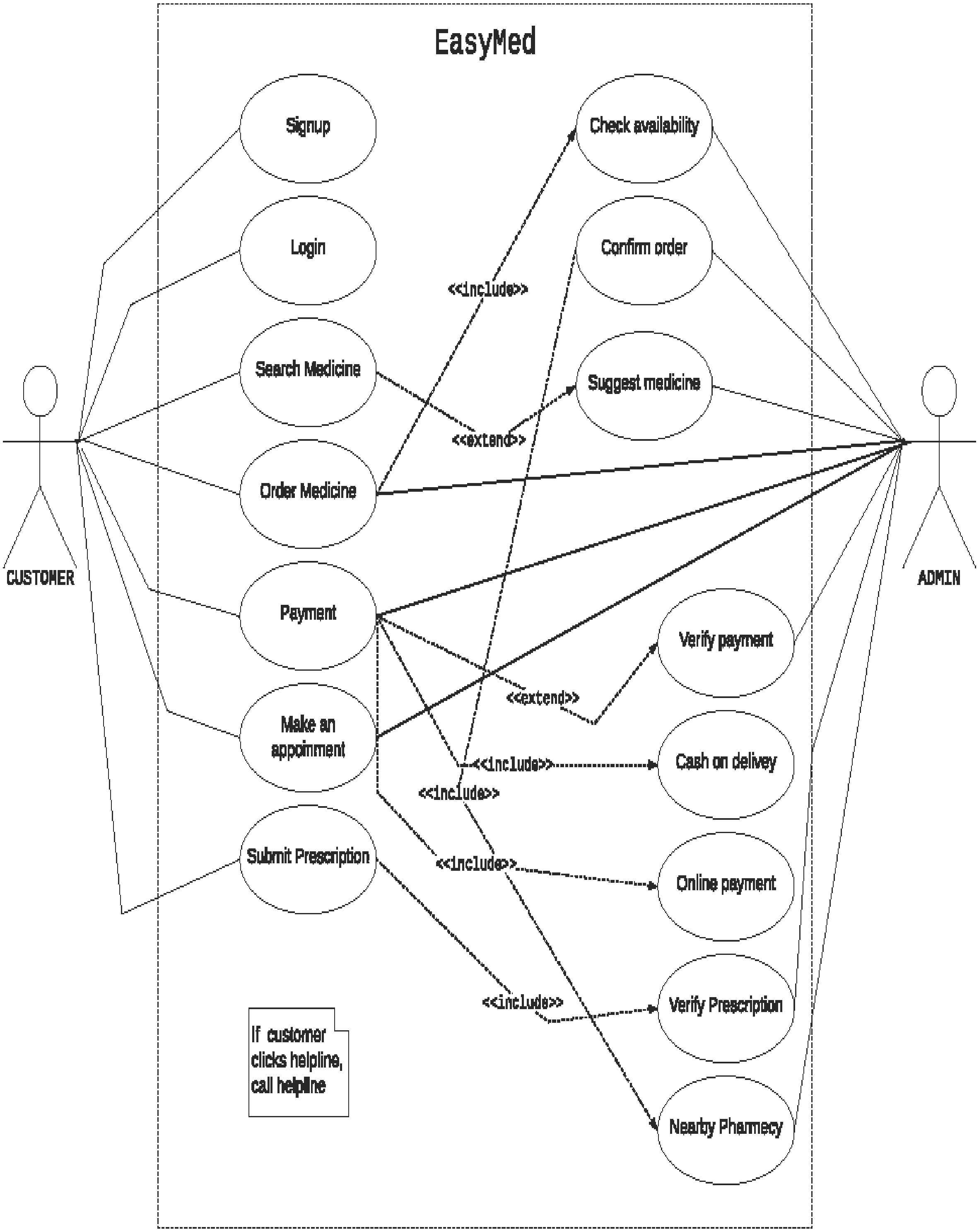


3.2 Database schema



3.3 UML diagrams

3.3.1 Use case diagrams



Use Case Name: Search Medicine

Actors: Customer

Preconditions: The user must know the name of the medicine.

Description ("Scenario"):

1. The user puts the name of the medicine or the name of the company in the search box.
2. The user is shown all the medicine with that name or sold by the company.
3. The user maybe shown medicine similar to the searched medicine.

Exceptions (Scenario):

The user puts the wrong name of the medicine or wrong name of the company.

Postconditions:

The user proceeds to select the medicine and add it to cart to confirm the order.

Use Case Name: Order medicine

Actors: Customer

Preconditions: The user must select the medicine he wants to buy .

Description ("Scenario"):

1. The user clicks on the medicine he wants to buy and add it's to cart.
2. The user clicks on the checkout option.
3. The user clicks on the payment option.

Exceptions (Scenario):

The user doesn't add the required medicine to cart.

Postconditions:

The user heads towards the payment option to complete his payment.

Use Case Name: Payment

Actors: Customer

Preconditions: The user must select how he wants to complete his payment.

Description ("Scenario"):

1. The user clicks on pay on delivery option to pay after receiving the medicine.
2. The user clicks on pay online option to pay through bkash.

Exceptions (Scenario):

The user doesn’t complete his payment through bkash then his order will be cancelled.

Postconditions:

The user gets a confirmation number through message after he finishes his payment using bkash.

Use Case Name: Make an appointment

Actors: Customer

Preconditions: The user need to select the the date and time he wants to make an appointment with the specialist to talk with him personally.

Description ("Scenario"):

1. The user clicks on make an appointment option.
2. The user puts his date and time of convenience to make an appointment.
3. The user may put emergency on his appointment request.

Exceptions (Scenario):

The user puts wrong date and time to make the appointment.

Postconditions:

The user gets a confirmation of the appointment with the specialist.

Use Case Name: Submit prescription.

Actors: Customer.

Preconditions: The user needs to submit a prescription to buy any sleeping medicine.

Description ("Scenario"):

1. The user selects submit your prescription option.
2. The user uploads the prescription in jpeg or pdf form.
3. The prescription is sent to the admin for verfiacation

Exceptions (Scenario):

Any false or modified document was submitted.

Postconditions:

The user gets to select the sleeping if his prescription was verified by the admin.

Use Case Name: Check availability.

Actors: Admin.

Preconditions: The admin checks if the required medicine is available in the nearby pharmacy.

Description ("Scenario"):

1. The admin sends a delivery man to the local pharmacy to check if the medicines are available in stock.
2. The admin confirms the order after finding the medicine in stock.

Exceptions (Scenario):

The medicine is not available in any pharmacy.

Postconditions:

The admin confirms the delivery man to complete the order.

Use Case Name: Confirm order.

Actors: Admin.

Preconditions: The admin verifies if all the process was done accordingly.

Description ("Scenario"):

1. The admin checks if the medicine is available in stock.
2. The admin confirms if the payment option was selected successfully.

Exceptions (Scenario):

Any of the steps had some error in it .

Postconditions:

The admin sends the customer a message of his order being confirmed.

Use Case Name: Verify payment.

Actors: Admin.

Preconditions: The admin verifies the payment which the customer has selected.

Description ("Scenario"):

1. The admin checks if the pay on delivery option was selected
2. The admin confirms if the payment through bkash was valid.

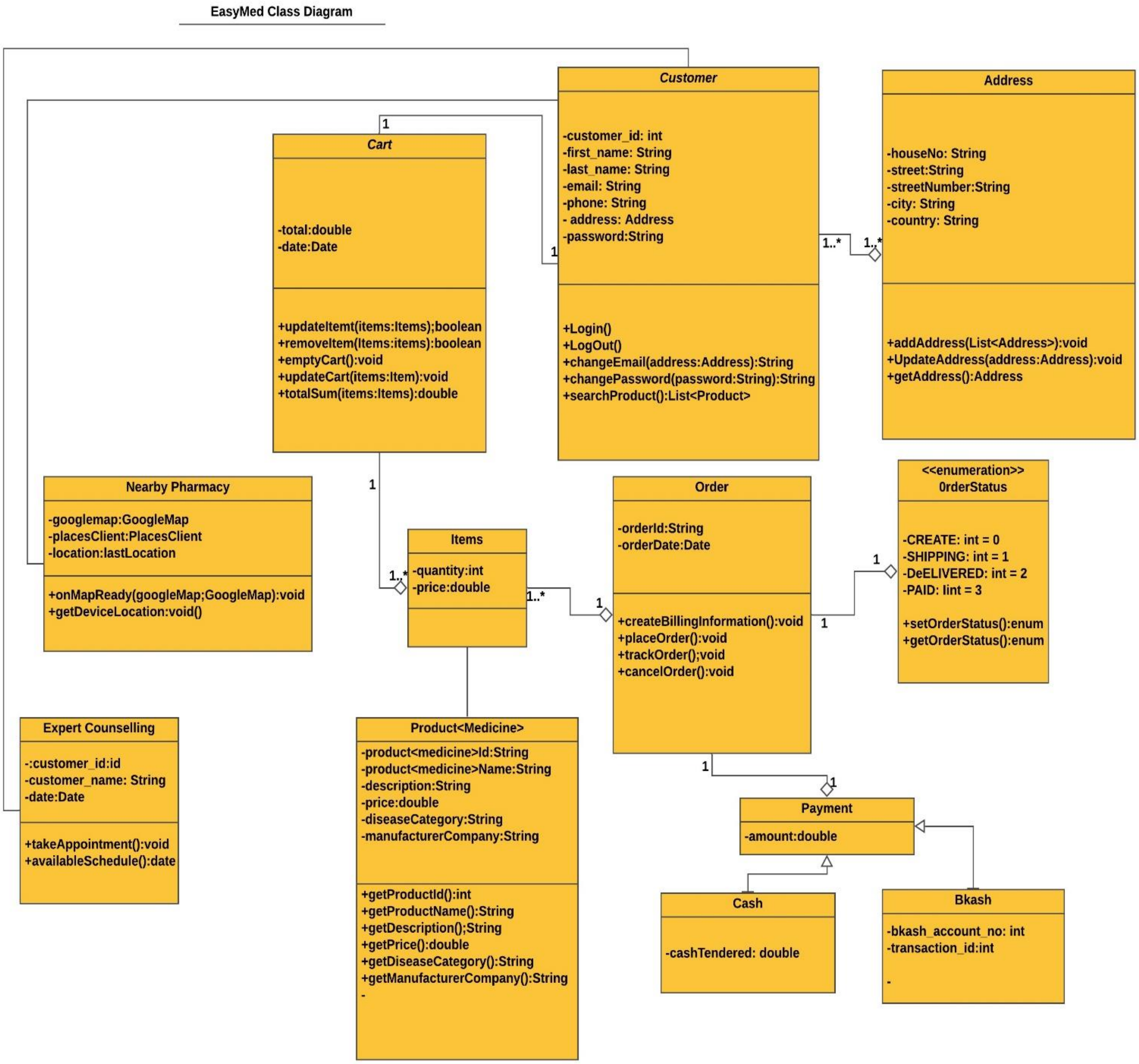
Exceptions (Scenario):

The payment through bkash was not received successfully.

Postconditions:

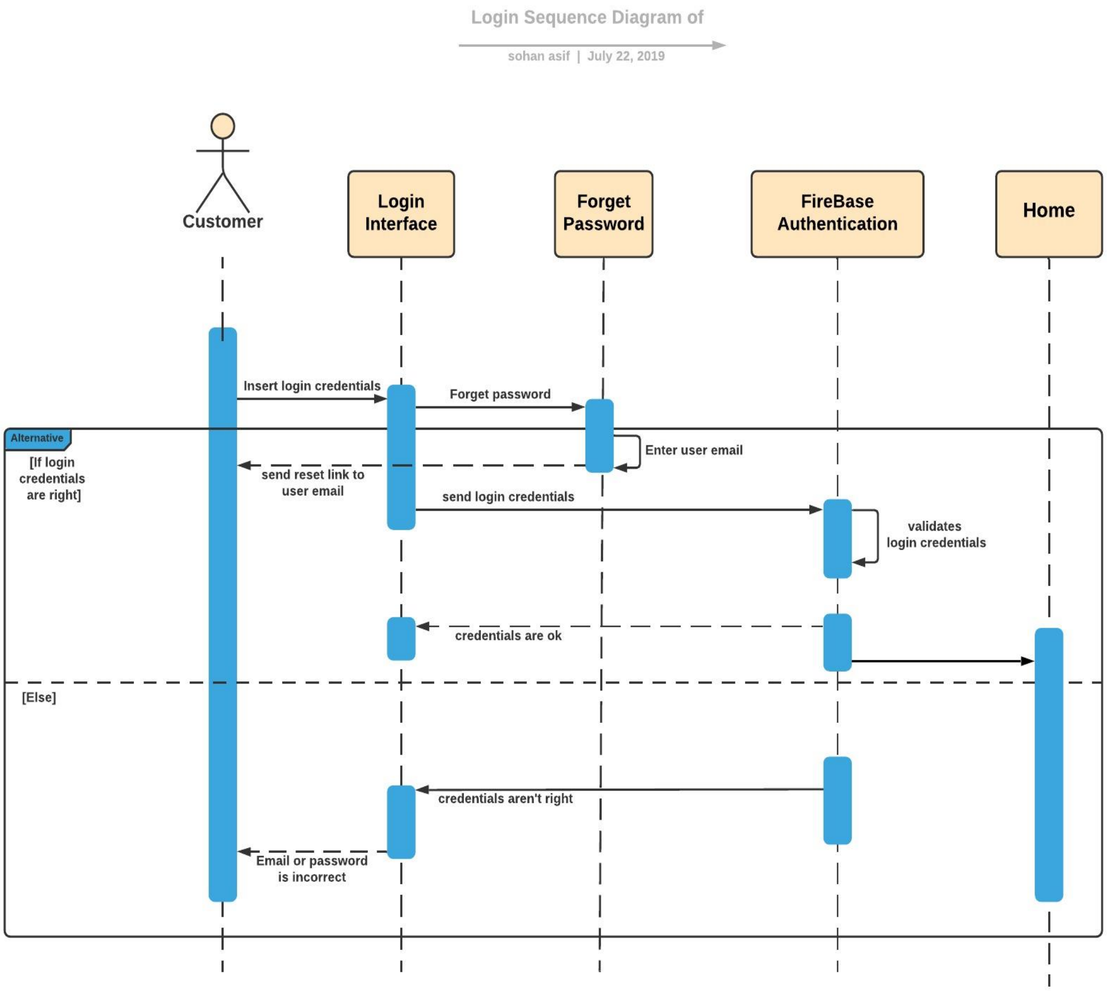
The admin sends the customer a message of his payment was received successfully.

3.3.2 Class diagrams

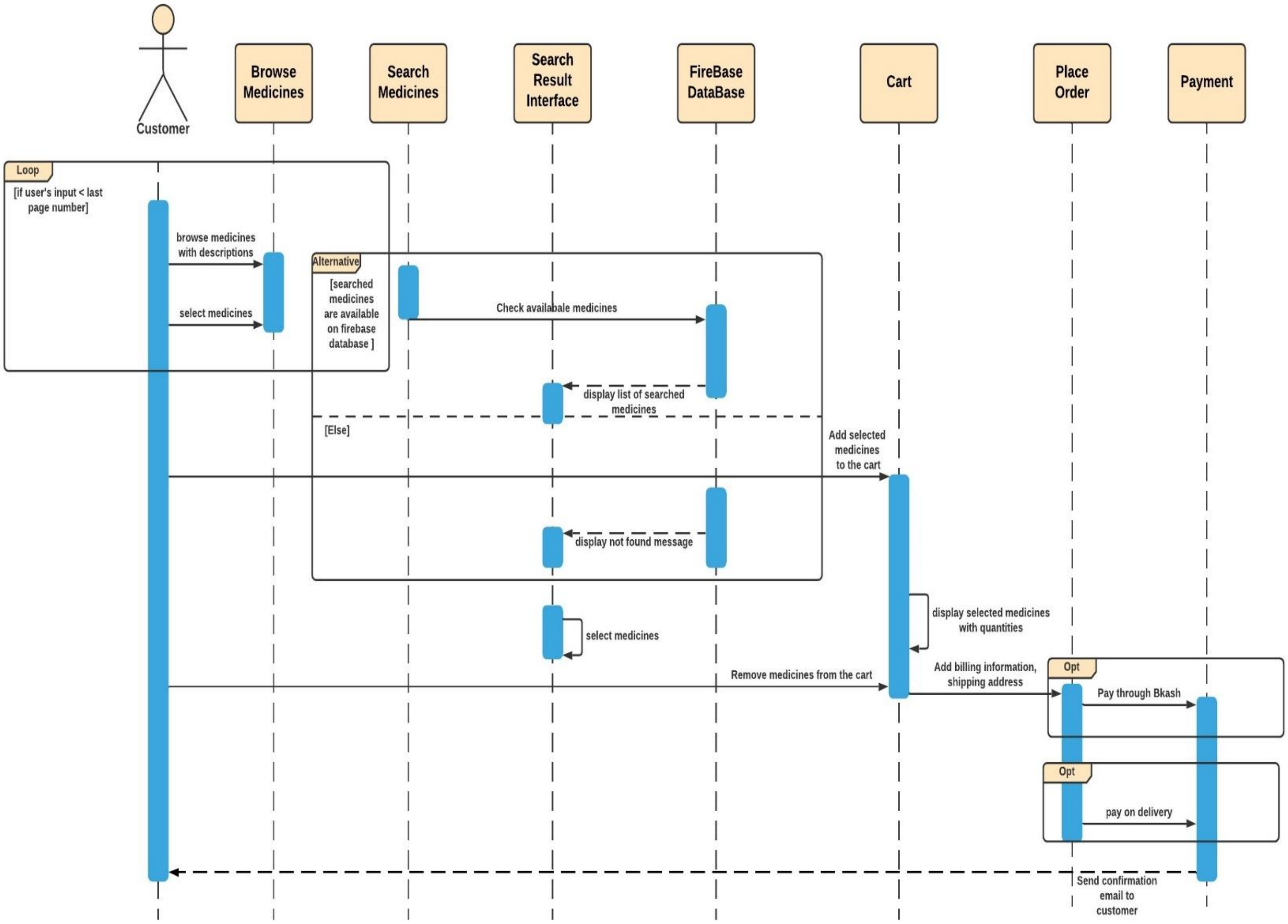


3.3.3 Sequence diagrams

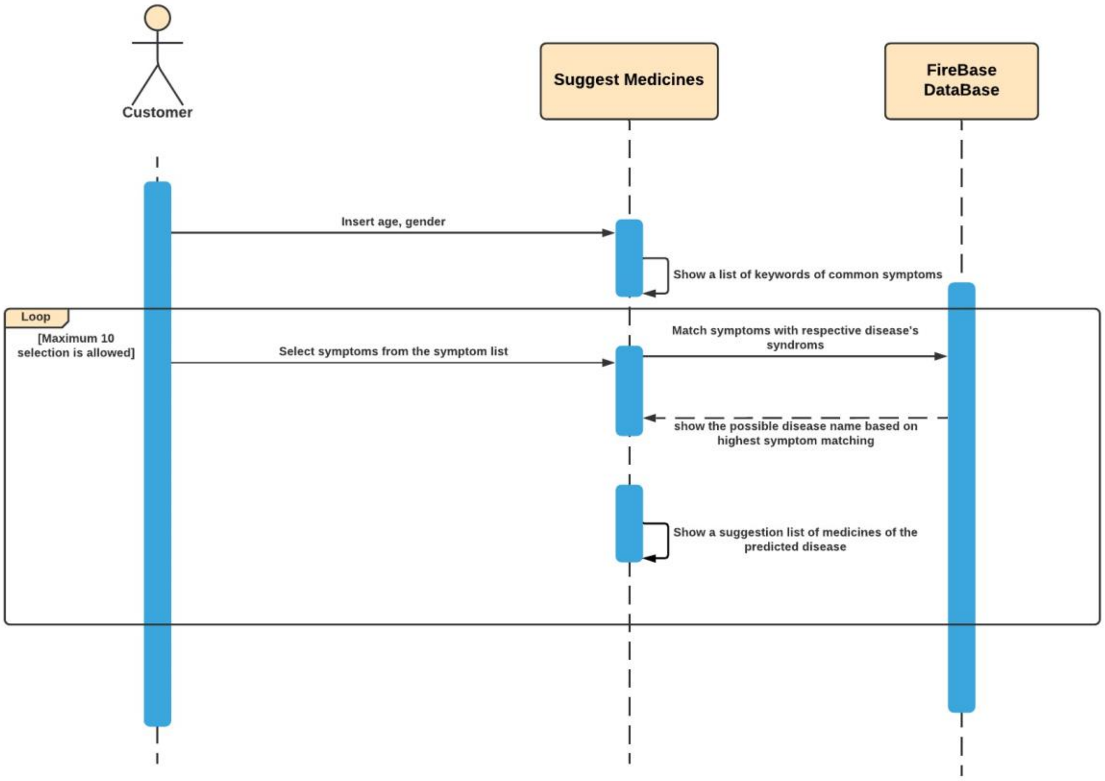
- Sequence diagram of login interface



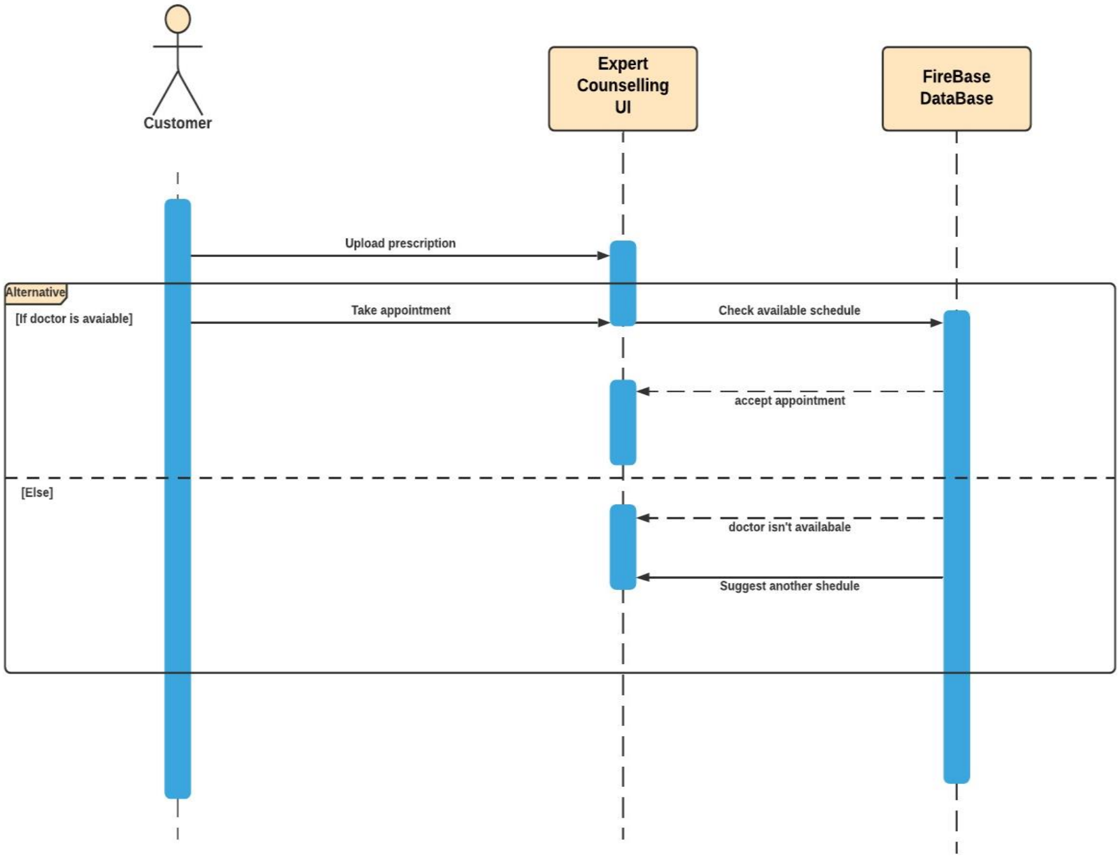
- Sequence Diagram of services of EasyMed


Customer
Browse Medicines
Search Medicines
Search Result Interface
FireBase DataBase
Cart
Place Order
Payment
Loop
[if user's input < last page number]
browse medicines with descriptions
select medicines
Alternative
[searched medicines are available on firebase database]
Check availabale medicines
display list of searched medicines
[Else]
Add selected medicines to the cart
display not found message
display selected medicines with quantities
select medicines
Remove medicines from the cart
Add billing information, shipping address
Opt
Pay through Bkash
Opt
pay on delivery
Send confirmation email to customer

- Sequence diagram of Suggest Medicines based on symptoms


Customer
Suggest Medicines
FireBase DataBase
Insert age, gender
Show a list of keywords of common symptoms
Loop
[Maximum 10 selection is allowed]
Select symptoms from the symptom list
Match symptoms with respective disease's syndroms
show the possible disease name based on highest symptom matching
Show a suggestion list of medicines of the predicted disease

- Sequence diagram of Expert Counselling service


Customer
Expert Counselling UI
FireBase DataBase
Upload prescription
Alternative
[If doctor is avaiable]
Take appointment
Check available schedule
accept appointment
[Else]
doctor isn't availabale
Suggest another shedule