



Wollo University

Kombolcha Institute of Technology

College of Informatics

Department of Software Engineering

Project Title: - Pharmacy Management System

Course on: Software Engineering Tools and Practices

PREPARED BY:

1. Solomon Seyoum.....20157/10
2. Getnet Begashaw.....1315/10
3. Yoseph Kasayee.....0636/10
4. Leul Eyasu.....1572/10
5. Motti Tulu.....0590/10
6. Kidist Araya.....1301/09

Advisor's name: - T.Bezawit

Kombolcha, Wollo, Ethiopia

May 2019

Acknowledgement

First of all, we would like to thank our God for the completion of this project proposal. Next we are extremely grateful our teacher as well as our advisor Ms. Bezawit Endriss for her constant support in all phases of the project. Finally, we would like to thank kombolcha pharmacy admin to give us the necessary information for this study.

Abstract

This project is concerned on drugs management system of **Web based pharmacy management system of kombolcha**. The web based pharmacy management system is a system that manages information about drug. The existing system uses manual system; and it results with data redundancy and inaccurate data storing, difficult to know the expiry date of drugs, difficult to manage or getting the exact place of the drug on the shelf easily, and no well arrangement of drugs. The experience of the manual system is used as an input for the newly proposed system. To solve this problem, we need to plan to develop a web based pharmacy management system to change the existing system from manual to web based application system. To do this system, we use different methodologies to get information about the system. We use different information gathering techniques such as Observation, interview, and software tools like dev. C++ to write the code, MS-word 2016 for writing documentation of the project and we use a database server MySQL, a web server to develop this system. We also use hardware tools like computer, USB flash to edit and store our data respectively.

Table of Contents

| | |
|--|----|
| Chapter One..... | 1 |
| 1. Introduction | 1 |
| 1.1 Background of the project..... | 1 |
| 1.2 Overview of the Existing System..... | 1 |
| 1.3 Statement of the problem | 2 |
| 1.4 Objective of the Project..... | 2 |
| 1.4.1 General Objective..... | 2 |
| 1.4.2 Specific Objective | 2 |
| 1.5 Feasibility Analysis | 3 |
| 1.5.1 Operational feasibility | 3 |
| 1.5.2 Technical Feasibility | 3 |
| 1.5.3 Economic feasibility | 3 |
| 1.5.4 Behavioral /Political feasibility..... | 3 |
| 1.6 Scope and Significance Project..... | 4 |
| 1.6.1 Scope of the Project..... | 4 |
| 1.6.2 Limitation of the Project | 4 |
| 1.6.3 Significance of the Project | 4 |
| 1.7 Methodology for the Project | 5 |
| 1.7.1 Data Gathering Methodology..... | 5 |
| 1.7.2 Development Tools | 5 |
| 1.7.2.1 Hardware Tools | 5 |
| 1.7.2.2 Software Tools | 6 |
| 1.7.3 System Analysis and Design Methodology | 7 |
| Chapter Two..... | 8 |
| System Analysis | 8 |
| 2.1 Introduction | 8 |
| 2.2 The Existing System Description..... | 8 |
| 2.3 Constraints..... | 9 |
| 2.4 Actors of the Existing System..... | 9 |
| 2.5 System Requirement Specification (SRS)..... | 10 |
| 2.5.1 Functional Requirements..... | 10 |

| | |
|---|----|
| 2.5.2 Nonfunctional Requirements | 12 |
| 2.6 Modeling Proposed Systems | 13 |
| 2.6.1 System Use Case Diagrams | 13 |
| 2.6.2 Use Case Documentation (for each use case identified) | 14 |
| 2.7 Domain Modeling with Class Responsibility Collaborator | 21 |
| 2.8 Sequence Diagram..... | 23 |
| 2.9 Activity Diagram..... | 28 |
| Chapter Three | 34 |
| System Design..... | 34 |
| 3.1 Introduction | 34 |
| 3.2 Goal of the Design..... | 34 |
| 3.3 The Purpose of Design | 35 |
| 3.4 Class Modeling Diagram..... | 35 |
| 3.5 Component Modeling | 36 |
| 3.6 Deployment modeling | 38 |
| Chapter Four..... | 39 |
| Coding | 39 |
| 4.1 Introduction | 39 |
| 4.2 Coding | 39 |
| 4.2.1 Forward engineering class diagram..... | 39 |
| 4.2.2 Reverse engineering | 43 |
| 4.3 Testing..... | 44 |
| 4.3.1 Unit Testing..... | 44 |
| Chapter Five | 46 |
| Conclusions and Recommendation | 46 |
| 5.1 Conclusions | 46 |
| 5.2 Recommendation..... | 46 |
| Références | 47 |

Table for Contents

| | |
|--|----|
| Table 1.1 Hardware tools | 6 |
| Table 1.2: software tools | 6 |
| Table 2.1: Functional requirement | 11 |
| Table 2.2: Nonfunctional requirement | 12 |
| Table 2.3-Use case description for login..... | 14 |
| Table 2.4: Use case description to manage drug..... | 16 |
| Table 2.5: Use case description for generate report | 17 |
| Table 2.6: Use case description for register patient | 18 |
| Table 2.7: Use case description for receive prescription | 19 |
| Table 2.8 Use case description for manage account | 20 |
| Table 2.9 Use case description for view comment..... | 21 |
| Table 2.10: CRC for Login | 22 |
| Table 2.11: CRC for Administrator..... | 21 |
| Table 4.1 unit testing..... | 45 |
| Table 2.12: CRC for Drug..... | 22 |

Contents for Figure

| | |
|---|----|
| Figure 2.1-Usecase Diagram | 13 |
| Figure 2.2: - sequence diagram for Login | 23 |
| Figure 2.3: sequence diagram for Generate Report..... | 24 |
| Figure 2.4: sequence diagram for register patient | 25 |
| Figure 2.5: sequence diagram for view comment | 26 |
| Figure 2.6: sequence diagram for manage account | 27 |
| Figure 2.7: sequence diagram for pay drug..... | 28 |
| Figure 2.8: activity diagram for login | 29 |
| Figure 2.9: activity diagram for generate report | 30 |
| Figure 2.10: Activity diagram for register patient..... | 31 |
| Figure 2.11: activity diagram for view comment | 32 |
| Figure 2.12: activity diagram for manage account..... | 33 |
| Figure 3.1: class diagram | 36 |
| Figure 3.2: Component diagram..... | 37 |
| Figure 3.3: Deployment diagram | 38 |
| 4.1 figure reverse engineering diagram..... | 43 |

Chapter One

1. Introduction

Pharmacy is a place where drugs, cosmetics and other medical related items are stored and sold for peoples. Our project focuses on the drug management system of a pharmacy. This project is concerned on drugs management system of Web based pharmacy management in case of kombolcha. The web based pharmacy management system is a system that manages information about drug. The current system is using manual system; it results with data redundancy and incorrect data storing, difficult to know the expiry date of drugs, difficult to manage or getting the exact place of the drug on the shelf easily, and no well arrangement of drugs. To overcome this problem, we need to plan a web based pharmacy management system to change the existing system from manual to web based application system.

1.1 Background of the project

Kombolcha pharmacy is one of the services in Amhara regional state. This organization was starting by giving small capacity in the drugs service. The aim of the organization is to provide efficient service to user to increase the capacity to provide drug for the recipient. From time to time the capacity of it is increasing but the organization was working with manual system. Since the organization is using manual system this makes the employee to fail with data redundancy erroneous data storing. Generally, this organization was established to save the life of the people who are affected disease by providing drugs. An Ethiopian person by drugs time to time, there is arising in the number of people who are affected by disease and the distribution of drugs become increasing. But the system is very old and paper based. Therefore, the team member interested to make it web-based system.

1.2 Overview of the Existing System

Kombolcha has its own drug managements system. The system is manual, which means there is no or little automation in the buying, storing and distribution process. currently this is done on trial and error approach, which means the drug store manager check whether the drugs are present or absent, expired or not. When the drug on the store becomes less the drug manager buys drug from well-known company. The manager also distributes these drugs to the dispensary then distributed these drugs to society.

1.3 Statement of the problem

In the existing system, most of the activities are performed manually. Because of this the current system has a lot of problems. This includes:

- It requires more resources: Consumes more resources and costs such as paper, pen and man power
- It is difficult to know the expiry date of drugs
- It is difficult to manage or getting quick information about drugs
- It is difficult to know the exact place of the drug on the shelf
- Data corruption due to time and place
- Searching data from a lot of paper is difficult
- It takes more time to retrieve, update recorded information
- It faces data inconsistency
- Percentage of security is less

1.4 Objective of the Project

1.4.1 General Objective

The main objective of this project is to develop web based pharmacy management system for Kombolcha society in order to provide efficient service for the system user.

1.4.2 Specific Objective: - The specific objectives of the project are:

- Understand the existing system.
- Collect and study requirement of the existing system.
- Model the new system using object oriented methodologies.
- Implementation of the project using programming language according to the design specification.
- Review the existing system to know the problem.
- To improve information sharing between the organization and the user.
- To provide easy way of commenting the system.
- Enable patient to see drugs from home by not going far apart.
- Enables patient the detailed information about each drug in their home.
- Finally implement and test the new system.
- Designing user friendly interface

1.5 Feasibility Analysis

Feasibility means answering questions relating to the utility and availability of the system that is going to be developed & it is the measure of how beneficial or practical of pharmacy drug information system will be to an organization. To get user acceptance and making the system easily understandable and accessible the new system considers the following feasibilities:

1.5.1 Operational feasibility

Our system is operationally feasible. The system operates in different devices which have internet access and by different users who have internet access, valid account type and verification code. Our system also is no more complex, it can be understood easily by the user. The user can access or operate the new system in anytime and anywhere. It contains various user-friendly interfaces that allow the user interacting with the system easily.

1.5.2 Technical Feasibility

In technical feasibility we should state that our new system can implement with current technology, a platform independent programming language such as Dev. C++ and with latest versions of software and also the user has enough knowledge and skill in using this technology. These development tools are most recent and popular technologies to develop online systems and design databases.

1.5.3 Economic feasibility

Economic feasibility determines the costs and benefits related to the developing project. So our project is economically feasible because this new project reduces the cost. When it is successfully completed it will use to store data & information for drugs & patients.

- Cost benefit Analysis
- Cost of the project

1.5.4 Behavioral /Political feasibility

The web based management system that we develop is politically feasible because this project will provide a lot of benefits for the society. It does not have a negative influence on life of the society and it will not interfere on the values and customs of the society.

1.6 Scope and Significance Project

1.6.1 Scope of the Project

The scope of the project is concentrating on how to manage the drug distribution and its inventory activities and applications of the drug in easy, efficient and functional way. It is also interested to analyze the system in relation with customer interaction. The scopes of this project include:

- View comment
- Manage drug
- View available drug
- View information
- Generate Report
- Send and receive prescription

1.6.2 Limitation of the Project

- Not support all nation and nationality of Ethiopians languages.
- The system cannot help blind people because of lacking of sound sensor machine.
- The system does not allow online drug shopping.
- The system requires internet connection.
- The system works only when database is active and the network is available.

1.6.3 Significance of the Project

After implementing this project, the organization will have a lot of benefits. These benefits are:

- Minimize time delay for getting information for drugs.
- It avoids redundancy of data.
- Displays drug information easily
- Less human power-it avoids labor cost for drug management cost by setting needed information from the system.
- It ensures the availability of documents in secured manner.
- Minimize employees' workload.

1.7 Methodology for the Project

1.7.1 Data Gathering Methodology

We gather the necessary information for our project from kombolcha pharmacy through the following methods:

Interview: This is one of data collection method that enables to gather information from the organization directly in the form of asking question and getting answers for those questions.

So, we use this method to gather information by asking the admin of the pharmacy some basic questions regarding the following issues during the interview: -

- How drug information management system is going on?
- During managing, are there any problems? If there, what are they?
- What requirements are needed for the process?
- Who is responsible for the drug management system?

Observation: - This is also another way of data collecting method. In fact, we have also used this observation method to gather data. This method enables us observing and understanding how drug information management is going on.

1.7.2 Development Tools

1.7.2.1 Hardware Tools

We use the hardware tool specifications listed below to develop our proposed system and for taking a backup. The hardware tools we used are listed by the following table:

| Hardware tools within their cost | | | | |
|----------------------------------|-------------------|----------|-----------------------|-----------------|
| No. | Name | Quantity | Unit price in Birr | Total |
| 1 | PC | 3 | 11000 | 33000.00 |
| 2 | Flash(16GB) | 2 | 250 | 500.00 |
| 3 | Paper | ½ pack | 0.50 | 100.00 |
| 4 | Pen | 3 | 5 | 15.00 |
| 5 | Printing cost | 150 | 1.00 | 150.00 |
| | Total cost | | | 33765.00 |

Table 1.1 Hardware tools

1.7.2.2 Software Tools

The software tools that we used to develop this project are shown by the following table.

| Software cost | |
|------------------------|--------------|
| Required software | Unit price |
| 1.Microsoft word 2016 | 40.00 |
| 2. Notepad++ | 30.00 |
| 3.JAVA net beans | — |
| 4.Enterprize architect | -- |
| Total cost | 70.00 |

Table 1.2: software tools

1.7.3 System Analysis and Design Methodology

We use the Object Oriented System Analysis and Design (OOSAD) Methodology for the development of our system instead of structured approach. We use waterfall approach methodology because it is very easy for software development and construction. It employs a top-down approach to projects, where the project has clearly defined steps from Requirement gathering to final implementation.

Because it is better way to construct and manage the objects that are implemented in our system and it is more acceptable due to its great advantage for Polymorphism, Abstraction, Encapsulation, Modularity, inheritance, Hierarchy, Concurrency, Persistence in terms of its:

- Increase reusability: - The object oriented provides opportunities for reuse through the concepts of inheritance, polymorphism, and encapsulation.
- Increased extensibility: - When we do changes on one part of the system such as addition of new features to the system, deletion of some part from the system and update records on the features it will be extensible to all part of the developed system.
- Flexibility: It is really flexible in terms of using implementations i.e., it is platform independent.
- Improved quality: - This methodology allows us to complete our system on time, on budget and meet the needed expectation of the users of our system with a high quality improved.
- Quality comes from increased participation of individual members of the group in the system development.

Chapter Two

System Analysis

2.1 Introduction

System analysis is finding out what happens in the existing system and deciding on what changes and the new features required and defining exactly what the proposed system must be. The process of system analysis is largely concerned with determining, developing and agreeing to the users' requirement. It also covers the major activities such as constructing a use case model, documenting the use case, constructing class diagrams, constructing sequence and activity diagrams, designing user interface prototyping about the proposed system. Even though this pharmacy gives many benefits to the customer, yet, it has its own problems.

2.2 The Existing System Description

- The task was performed manually: - patients, drugs and other related information are stored in the paper.
- There is no well arrangement of drugs: - drugs are not arranged in a good manner because of huge amount of drug stored in one store office.
- It's difficult to know the expiry date of drugs: - there are many different types and number of drug in one office because of these it is difficult to know the expiry date of each and every drug.
- It's difficult to know the Drug that has less quantity: - there are so many different numbers of drugs in the pharmacy so it is difficult to know which drug is less quantity in the pharmacy.
- It is difficult to manage or getting quick information: - because of information about drugs, patients and other medical related information are stored manually; it is difficult to search and get the needed information.
- Security: - All the records associated with the manual system are recorded and stored manually. The security that the system provides for the privacy of those records is not in good manner. The system shouldn't provide sufficient protection for access and manipulation of the records associated with the system.

2.3 Constraints

There are two kinds of risk assessment in any software development life cycles. Those are Development process risks and Product risk. While developing this system, the project team may encounter different types of risks Such as:

- Skill constraints: - we have a skill constraint because we have no enough knowledge on programming languages.
- Resource: - unavailability of some resources will also under consideration.
- Internet connection: -The unavailability of data source (such as shortage of internet connection on time) may extend the project completion time.
- Lack of enough reference books: - We can manage this problem by using other methods like internet but there is no more available Connection Service.
- Power fluctuation problem: - It is using laptop that have high power backups are used.
- Damage computers within viruses: - that it is managed by using backup.
- Shortage of time: We managed such problem by using additional time from our rest time
- Virus: virus can attack our project: - We used updated antivirus to manage this problem.

2.4 Actors of the Existing System

This illustrate the actors involving the current system, those actors in the existing systems are:

Pharmacist:-

- Distribute the drugs of the dispensary.
- Manage drugs of the existing system.
- Order drugs from different organization.

Customer: -

- Ordered drugs as they want.
- Ask information.
- Asking price of the drugs.
- Pay the cost of the drugs.

Administrator: -

- Send prescription paper to the dispensary.
- Is the one who maintains the system and manage account.
- Virus: virus can attack our project: - We used updated antivirus to manage this problem.

2.5 System Requirement Specification (SRS)

A requirement is a feature that the system must have or a constraint that it must satisfy to be accepted by the pharmacy system. It determines the needs of everyone who will be the user of the proposed system of our project users such as: administrator, patient, pharmacist, etc. Generally, the requirement of the new system can be viewed as:

2.5.1 Functional Requirements

The functional requirement of the Project is defining a function of our system and its components. A function is described as a set of inputs, the behavior, and outputs. Functional requirements also describe the relations between the system and the user or the environment. Here, the 'relations' means the direct or indirect interactions between the user and the system. The functional requirement of our system is shown below within the following table:

| R. Id | Requirement | Category |
|-------|--|----------------------|
| R1 | The system should allow employee login to the system by using their user name and password. | Login |
| R2 | The system should allow admin to see reports that are done on the daily, weekly, monthly and yearly Receipt and other related operation. | Checking Report |
| R3 | The system should allow the admin to create account for employees | Create account |
| R4 | The system must allow system admin to update accounts. | Update account |
| R5 | The system must allow employees to update their account. | Update account |
| R6 | The system must allow the system admin to delete user account. | Delete account |
| R7 | The system must allow the system admin to register and add employee. | Register and add |
| R8 | The system must allow pharmacist to update drug. | Update drug |
| R9 | The system must allow the pharmacist to delete expire drugs. | Delete expire drug |
| R10 | The system must allow the pharmacist to generate report. | Generate report |
| R11 | The system must allow the admin to get comment about the system. | Getting comment |
| R12 | The system must allow the pharmacist to generate receipt. | Generate receipt |
| R13 | The system must allow the admin, patient, pharmacist to logout. | Logout |
| R14 | The system Allows authorized user /person to update data in database. | Update Database |
| R15 | The system must allow pharmacist to arrange the type of drug based on pharmacological action on the shelf. | Arrange type of drug |
| R16 | The system must allow the patient to receive prescription. | Receive prescription |
| R17 | The system should allow the administrator to register patient. | Register patient |

Table 2.1: Functional requirement

2.5.2 Nonfunctional Requirements

The non-functional requirement of the system deals with how well the system provide service to the user. The non-functional requirement of our system is shown below within the following table:

| Req.ID | Requirement | Category |
|--------|---|----------------|
| R1 | All pages should be supported with English languages. | Usability |
| R2 | The system shall contain strong password to hack. | Reliability |
| R3 | The system shall have simple user interface. | Usability |
| R4 | The system shall support one user per minute. | Performance |
| R5 | The system will use low utilization of system resource in terms of space and time | Performance |
| R6 | The system administrator can able to update, modify, delete or access patient data. | Integrity |
| R7 | The system shall not fail at all | Reliability |
| R8 | The system must be display or respond an error message. | Error handling |
| R9 | The system shall run on Linux/windows. | Portability |

Table 2.2: Nonfunctional requirement

2.6 Modeling Proposed Systems

2.6.1 System Use Case Diagrams

System use case diagram depicts a collection of use cases, actors, their associations, and a system boundary box. A use case describes a sequence of actions that provide a measurable value to an actor and is drawn as a horizontal ellipse. An actor is a person, organization, or external system that plays a role in one or more interactions with our system (actors are drawn as a stick figures). It also shows use case reusability by including <<include>> and <<extend>> relationships between use cases. so we can represent a use case diagram with the following figure:

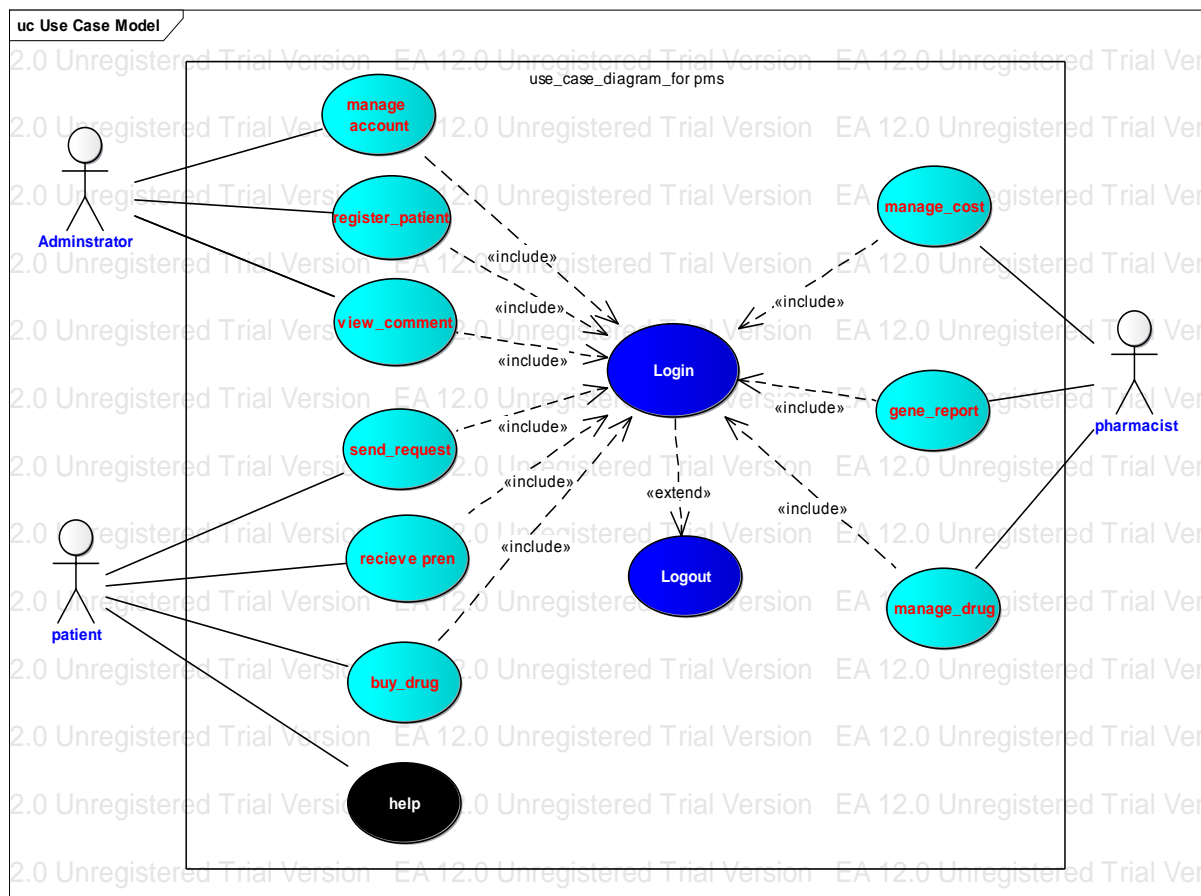


Figure 2.1-Usecase Diagram

2.6.2 Use Case Documentation (for each use case identified)

The next step is to document each of the above use case events to determine the requirement use Cases as described in the following table one by one.

| | |
|------------------------------|---|
| Use case name | Login |
| No. | UC1 |
| Actor | Administrator, patient, pharmacist |
| Description | To provide privileges for actors/users those are login to system in order to perform different tasks. |
| Pre-condition | All users of the system must have a valid account. i.e. valid user name and password |
| Basic course of action | <ol style="list-style-type: none">1. Open home page2. User requests to see Login page3. The system displays the login interface4. User fills his/her user name and password and select user type.5. The user requests to login/submit6. The system verifies the entered username and password.7. System display interface of appropriate page8. The use case end |
| Alternate courses of actions | If there is an error during filling the form the system displays an error message, and then back to basic course of action 1. |

Table 2.3-Use case description for login

| | |
|---|--|
| Use case name | Manage drug |
| No. | UC2 |
| Actor | Pharmacist |
| Description | It helps pharmacist to manage drug |
| Precondition | Pharmacist login first. |
| Post condition | Manage drugs |
| Basic course of action | |
| Actor's action | System response |
| 1.request to manage drug 3. Click on manage drug. 5. use case end | 2. The system displays the page with its option. <ul style="list-style-type: none">➤ Order drug➤ View drug➤ Register drug➤ Update drug➤ Delete drug 4. The system displays the drug from database |
| Alternate course of action | if there is no drug the system returns back to basic course of action 2. |

Table 2.4: Use case description to manage drug

| | |
|---|--|
| Use case name | Generate report |
| No. | UC3 |
| Actor | Pharmacist |
| Description | It helps the pharmacist to generate report. |
| Precondition | The pharmacist login. |
| Post condition | Pharmacist generates report. |
| Basic course of action | |
| Actor action | System response |
| 1. Request to generate report. 3. The pharmacist fills the form. 4. The pharmacist clicks report button 7. Use case end. | 2. The system displays report form. 5. the system verifies the outdate value 6. the system displays report send successfully |
| Alternative course of action | if the value is not valid returns to Basic course action of 3. |

Table 2.5: Use case description for generate report

| | |
|--|--|
| Use case name | Register patient |
| No. | UC4 |
| Actor | Administrator |
| Description | This use case allows admin to register patient. |
| Precondition | Administrator login first |
| Post condition | Register patient |
| Basic course of action | |
| Actor's action | System response |
| 1. The Administrator request to register patient 3.The Administrator fills the form. 4 Click on register button. | 2. The system displays the registration form. 5. The system verifies the data. 6. The system display registered patient. 7. Use case end. |
| Alternate course of Action | If the data that the pharmacist fills is not valid back to Basic course action of 3. |

Table 2.6: Use case description for register patient

| | |
|---|--|
| Use case name | Receive prescription |
| No. | UC5 |
| Actor | Patient |
| Description | The patient Receive prescription from the pharmacist |
| Precondition | Patient should login the system |
| Post condition | The Patient receive prescription from pharmacist |
| Basic course of action | |
| Actor's action | System response |
| 1. Request to receive prescription. 3. Click receive button. 5. Use case end. | 2. The system displays prescription page. 4. The system display received prescription successfully. |
| Alternate course of action | If no prescription the system returns to basic course of action 1. |

Table 2.7: Use case description for receive prescription

| | | |
|------------------------|---|---|
| Use case Id | Use case no | |
| Use case name | Manage account | |
| Actor | Admin | |
| Precondition | The system Administrator must login to the account | |
| Description | This activates is performed when the admin manage account | |
| Goal | To control the system worker | |
| Basic course of action | Actor action | System response |
| | Step1: Admin enter user name and password Step4: Admin select 1.create account 2.upfate account 3.delete account If create account Step6:Admin enter user account information | Step2: the system checks the authentication of username and password Step3: the system displays admin page Step 5: System display create account page. Step7: System check create user account information. Step 8: System creates user |

| | | |
|------------------------------|--|---------|
| | | account |
| Post condition | The system admin successfully create, update and delete the account | |
| Alternative course of action | A: Invalid information entry. 1. The system displays error message 2. Go to S6 fill again | |
| Exit condition | Logout | |

Table 2.8 Use case description for manage account

| | | |
|------------------------|---|----------------------------|
| Use case Id | Use case no | |
| Use case name | Comment the system | |
| Actor | Store manager, pharmacist ,patients | |
| Precondition | User must have valid email address to comment. | |
| Description | Comments the pharmacy system About anything | |
| Goal | To give the weakness and the strength of the system | |
| Basic course of action | Actor action | System response |
| | Step2: user select comment | Step1 system displays user |

| | | |
|------------------------------|--|--|
| | the system link Step4: user write comments about the system | page: Step3: system displays comments' form Step 5: system check comment information. Step 6:system displays comment submitted. |
| Post condition | User send comments to the system | |
| Alternative course of action | A: fail to send the comment 1. the system displays error message. 2. got to step 4 to fill again comment validly | |
| Exit condition | Logout | |

Table 2.9 Use case description for view comment

2.7 Domain Modeling with Class Responsibility Collaborator

Class Responsibility Collaborator (CRC) Modeling is a collection of standard index cards that have been divided into three sections.

- On top of the card, the class name.
- On the left, the responsibilities of the class.
- On the right, collaborators with which this class interacts with Admin.

| Login | |
|-----------|----------------|
| User name | administrator, |
| Password | patient, |
| | pharmacist |
| login () | |
| cancel() | |

Table 2.10: CRC for Login

| Administrator | |
|-------------------------|------------|
| Name | Pharmacist |
| Address | Patient |
| Phone number | |
| Sex | |
| Age | |
| Send prescription () | |
| Receive prescription () | |

Table 2.11: CRC for Administrator

| Drug | |
|-------------------|------------|
| Drug Name | pharmacist |
| Drug id | |
| Expire date | |
| Manufactured date | |
| Register drug () | |
| Update drug () | |
| Delete drug () | |

Table 2.12: CRC for Drug

2.8 Sequence Diagram

A sequence diagram models how the classes of objects interact with each other over time as the system runs. So, it will be prepared for each use case to show how different objects interact with each other to achieve the functionality of the use case as follows:

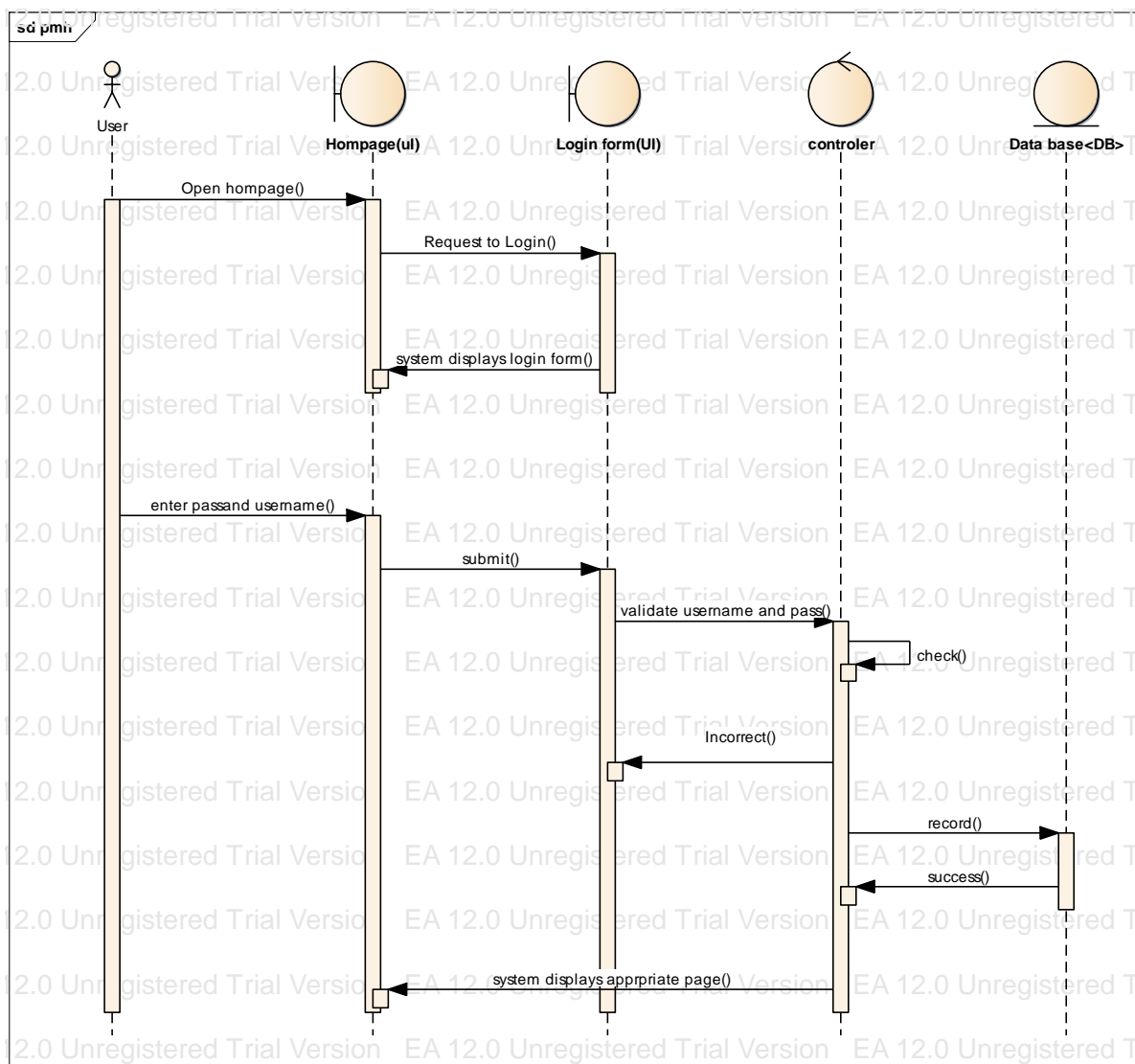


Figure 2.2: - sequence diagram for Login

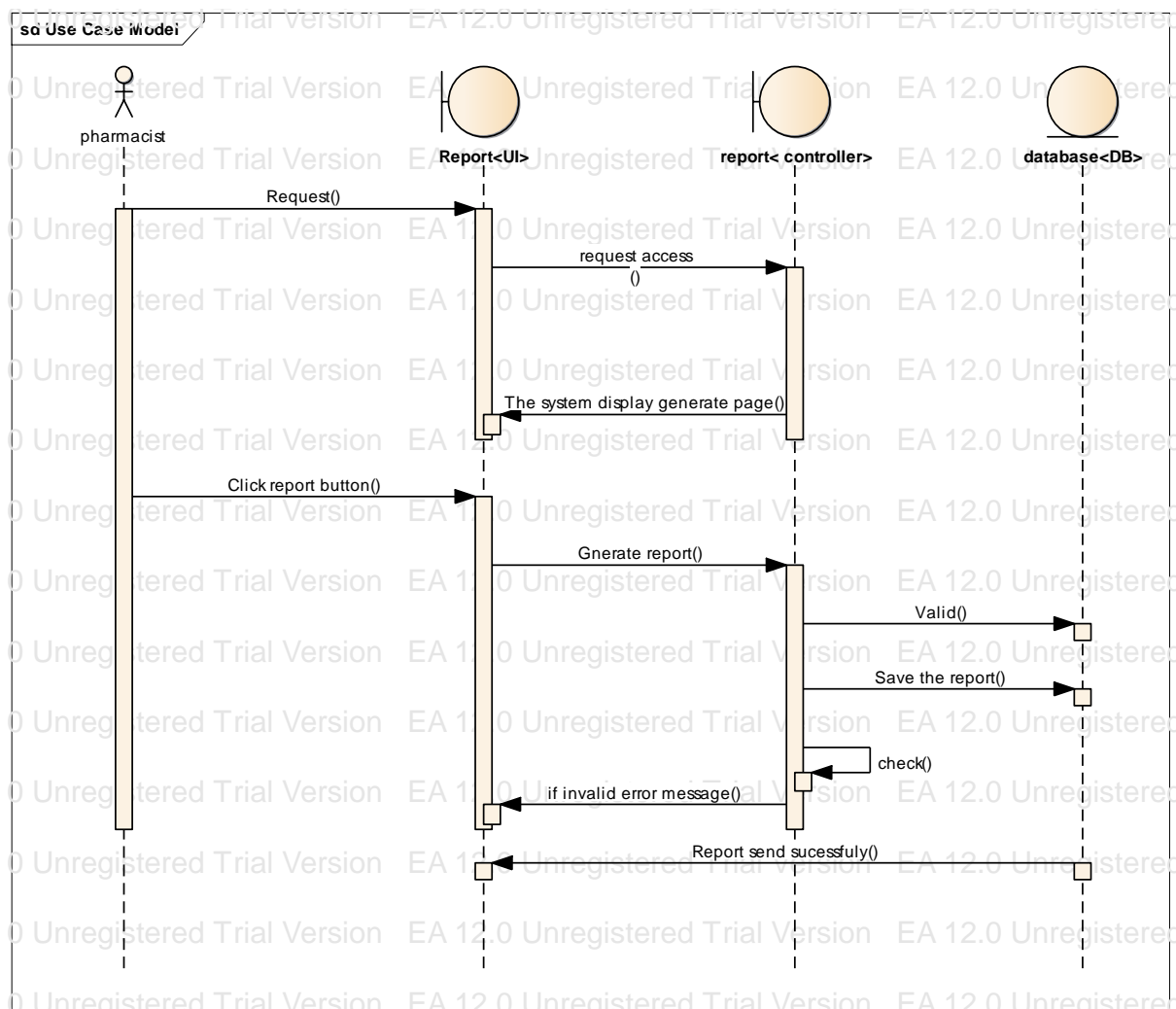


Figure 2.3: sequence diagram for Generate Report

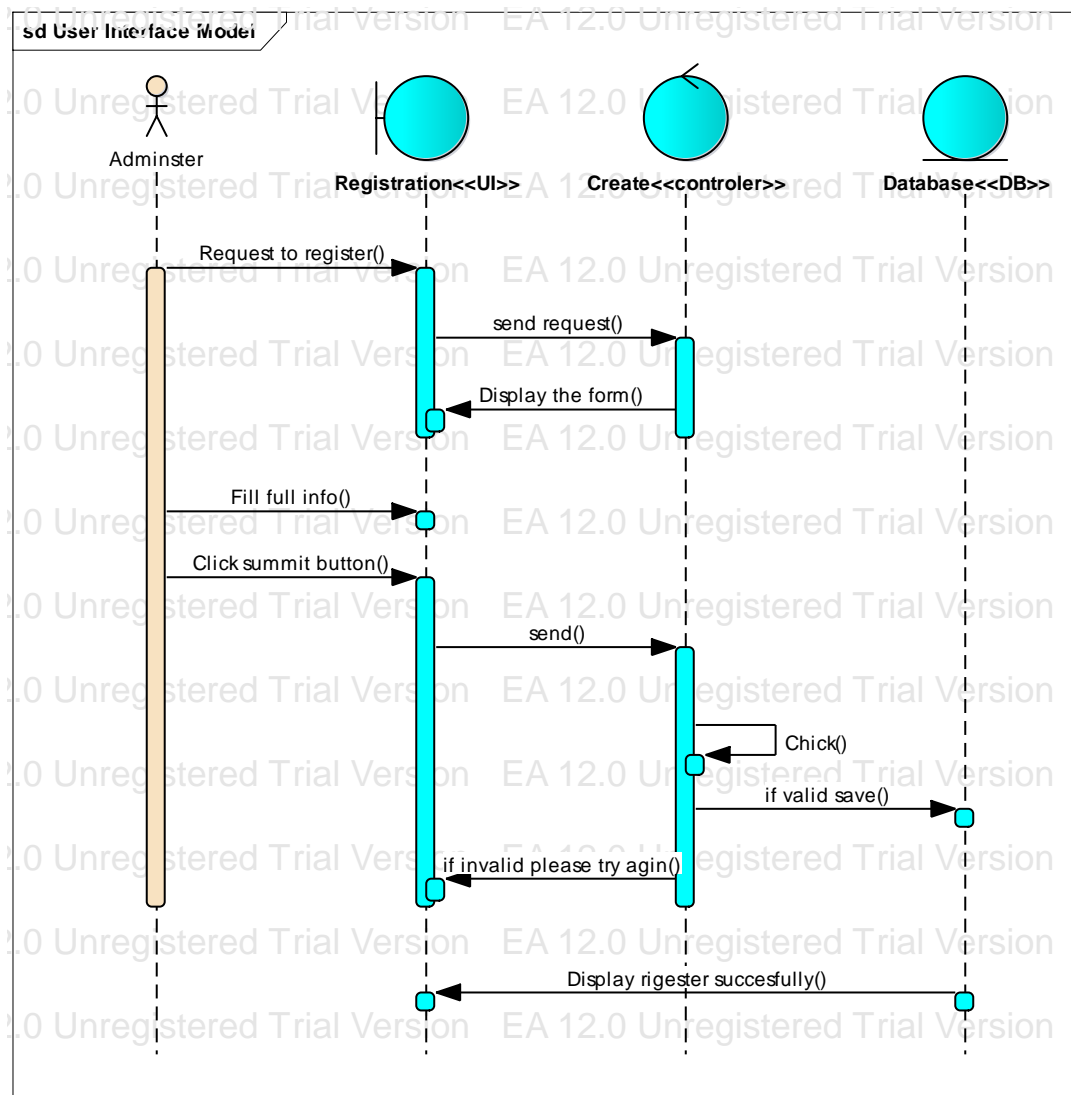


Figure 2.4: sequence diagram for register patient

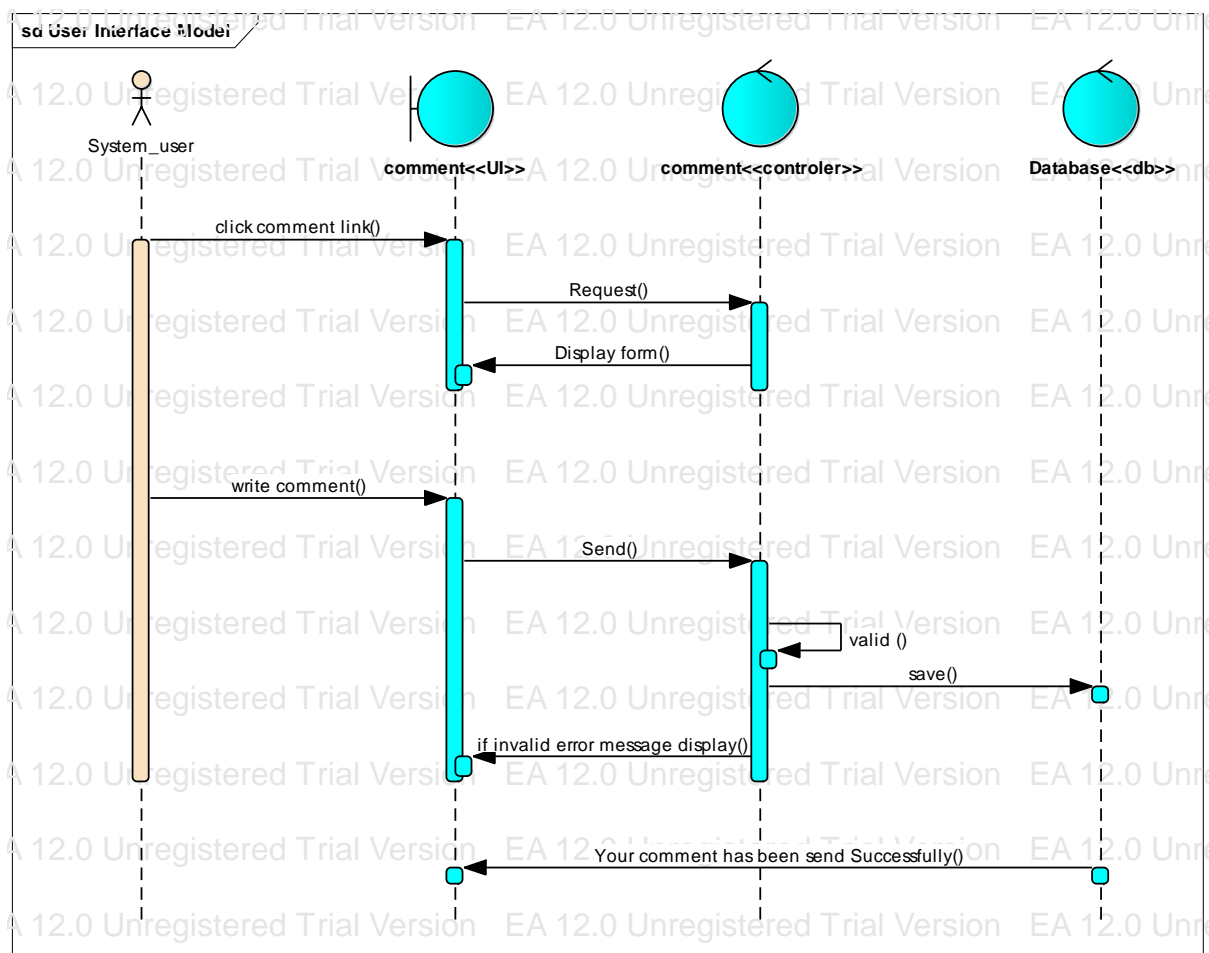


Figure 2.5: sequence diagram for view comment

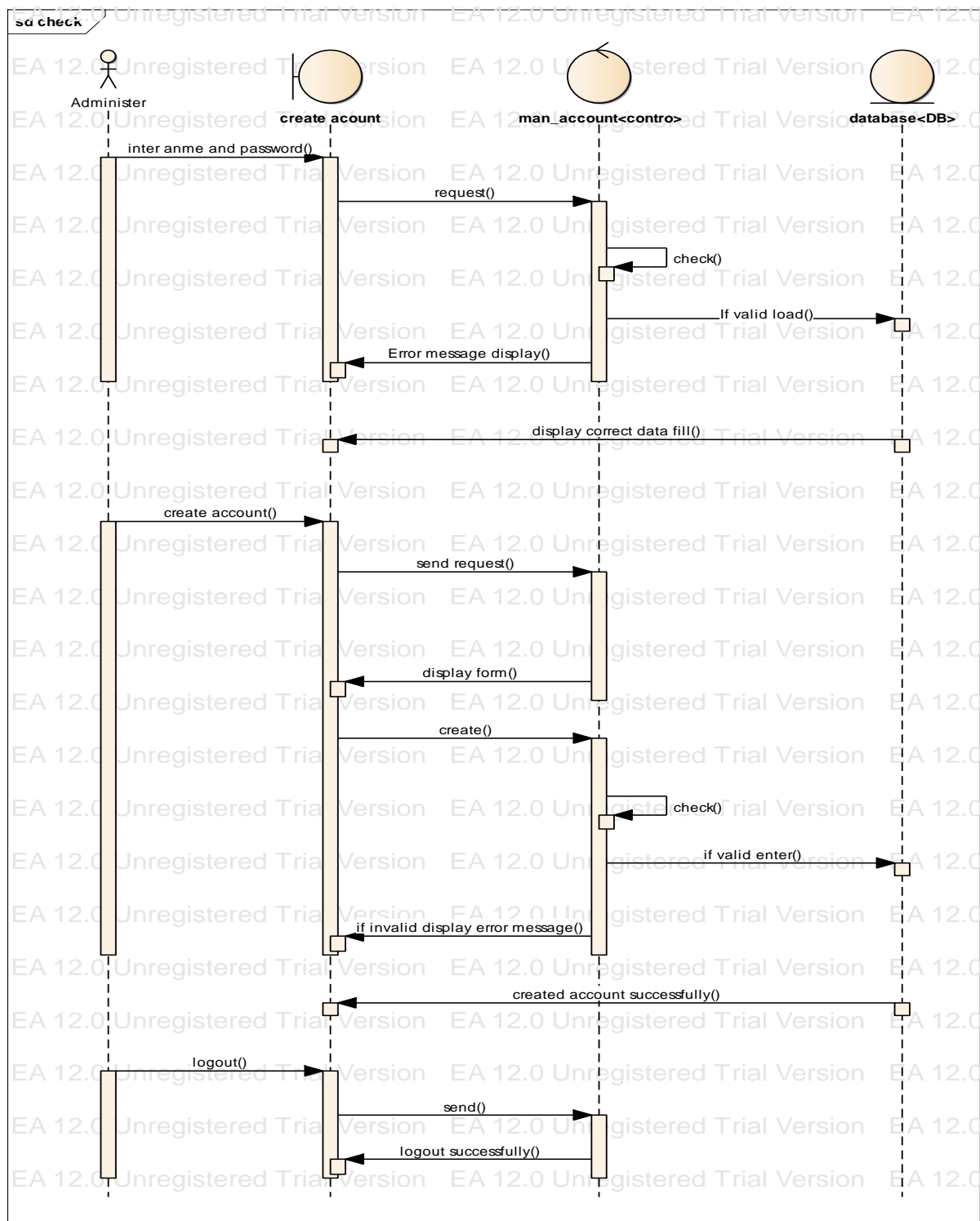


Figure 2.6: sequence diagram for manage account

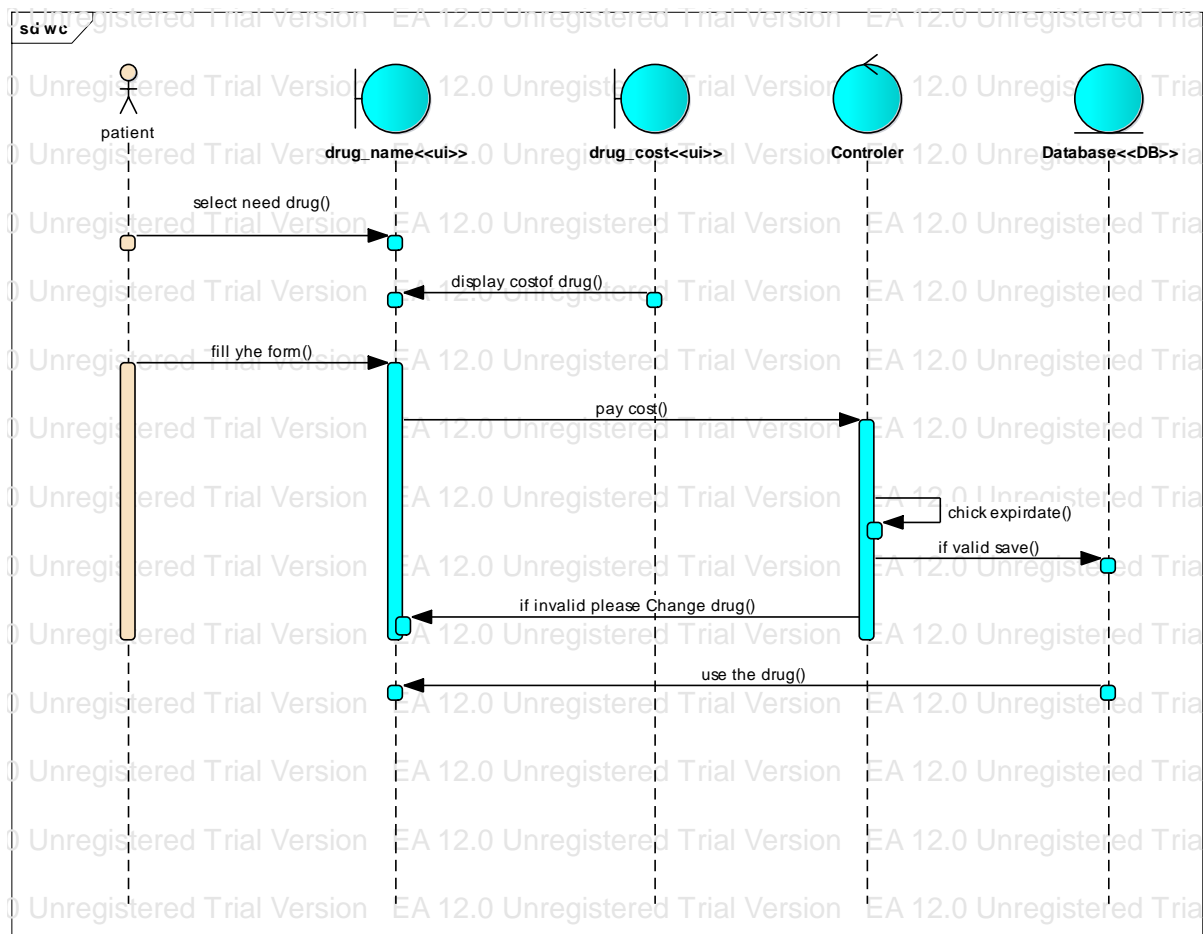


Figure 2.7: sequence diagram for pay drug

2.9 Activity Diagram

An activity diagram is essentially a flowchart, showing flow of control from activity. It involves:

- Modeling the sequential (and possibly concurrent) steps in a computational process
- Modeling the flow of an object as it moves from state to state at different points in the flow of control. So, we now draw an activity diagrams for each use case to show the operations/activities performed by use cases to achieve their functionality:

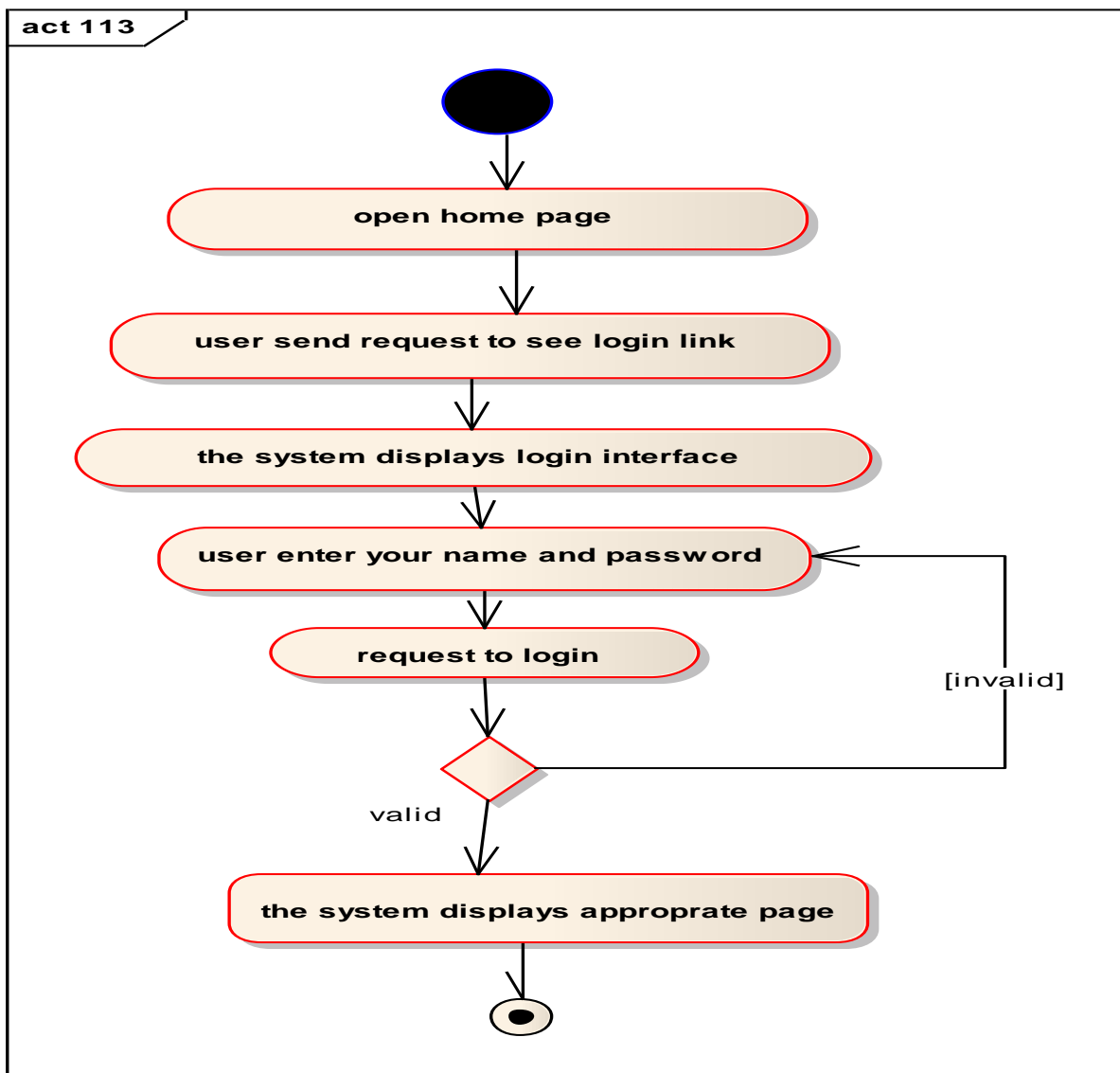


Figure 2.8: activity diagram for login

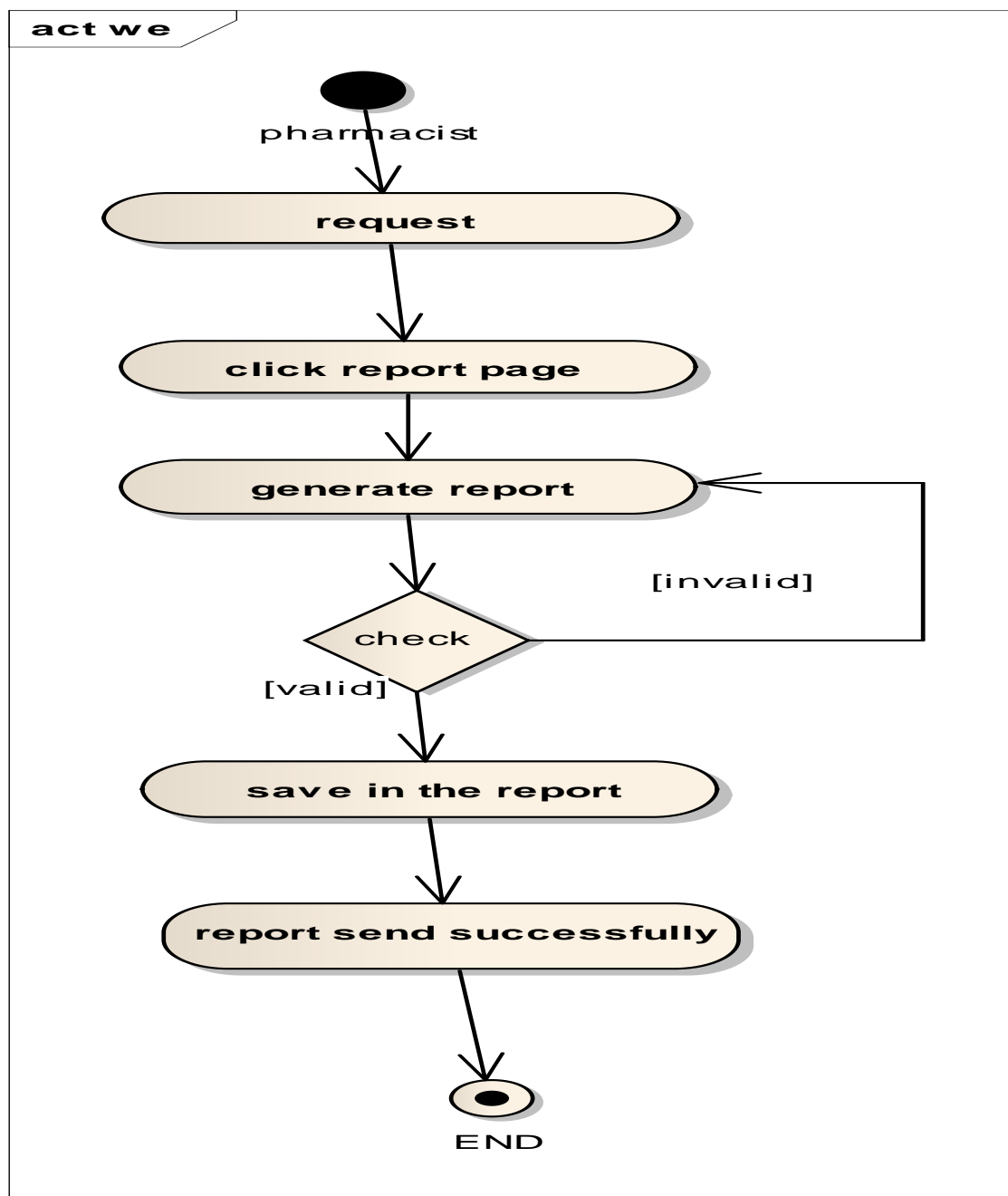


Figure 2.9: activity diagram for generate report

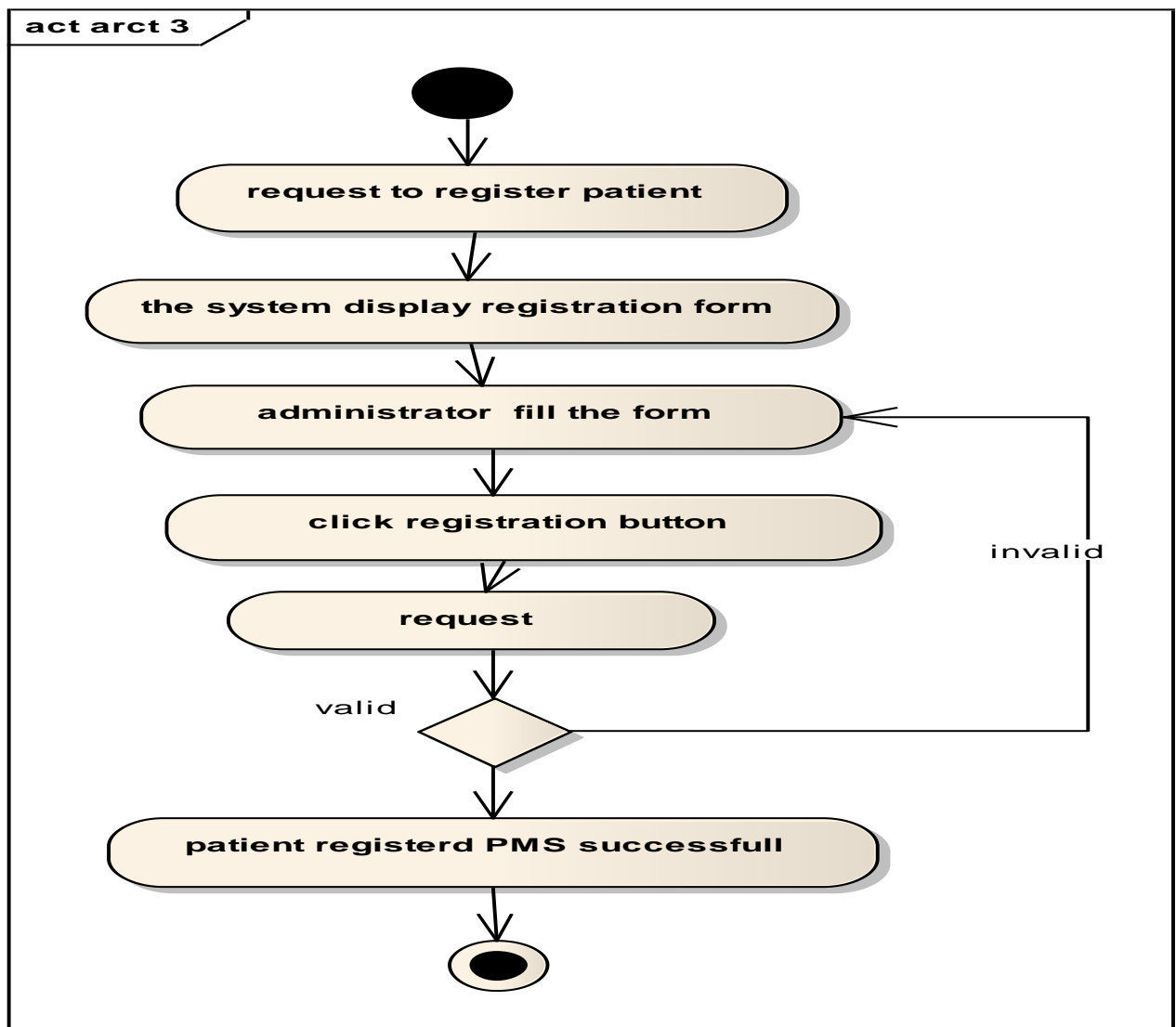


Figure 2.10: Activity diagram for register patient

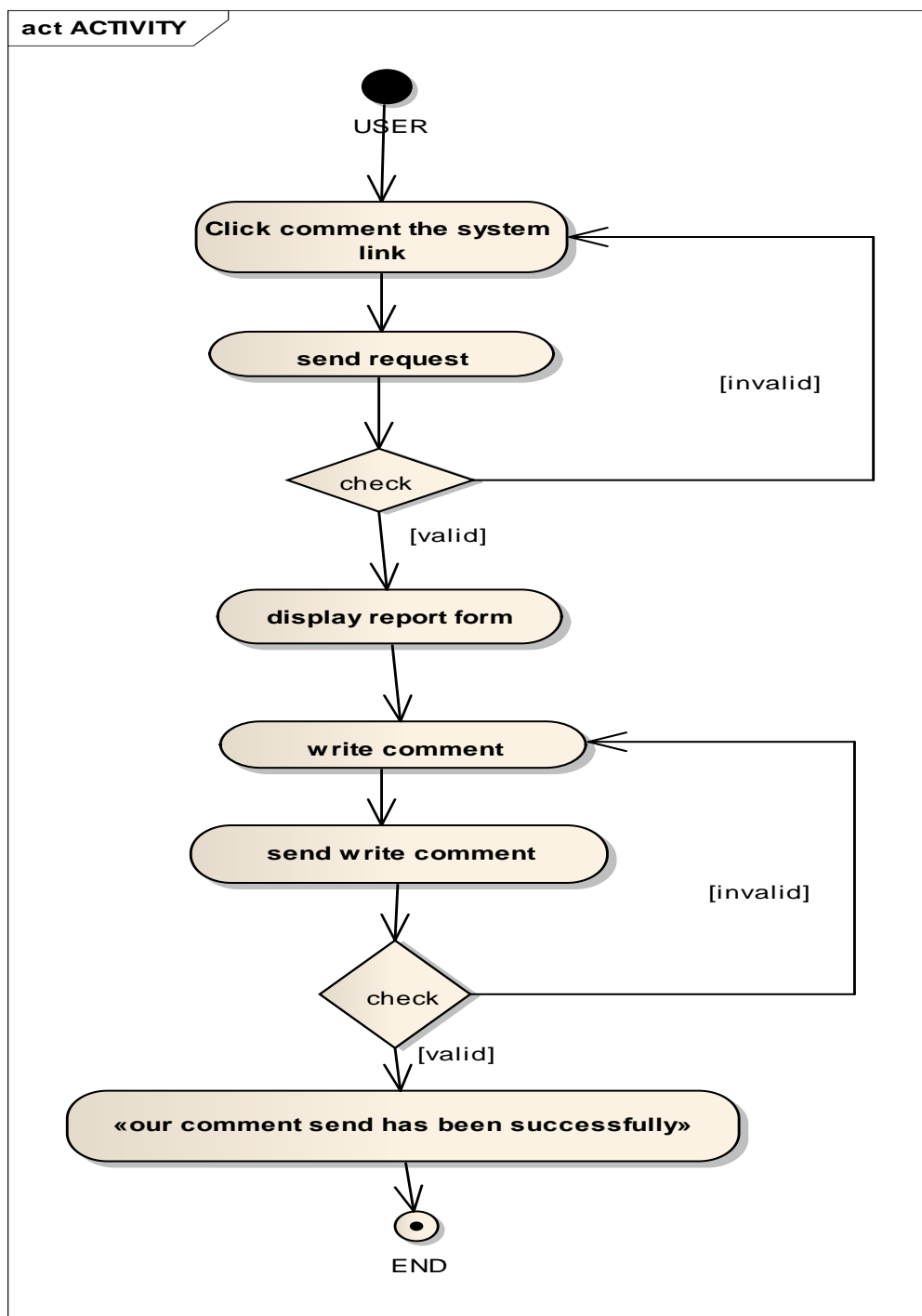


Figure 2.11: activity diagram for view comment

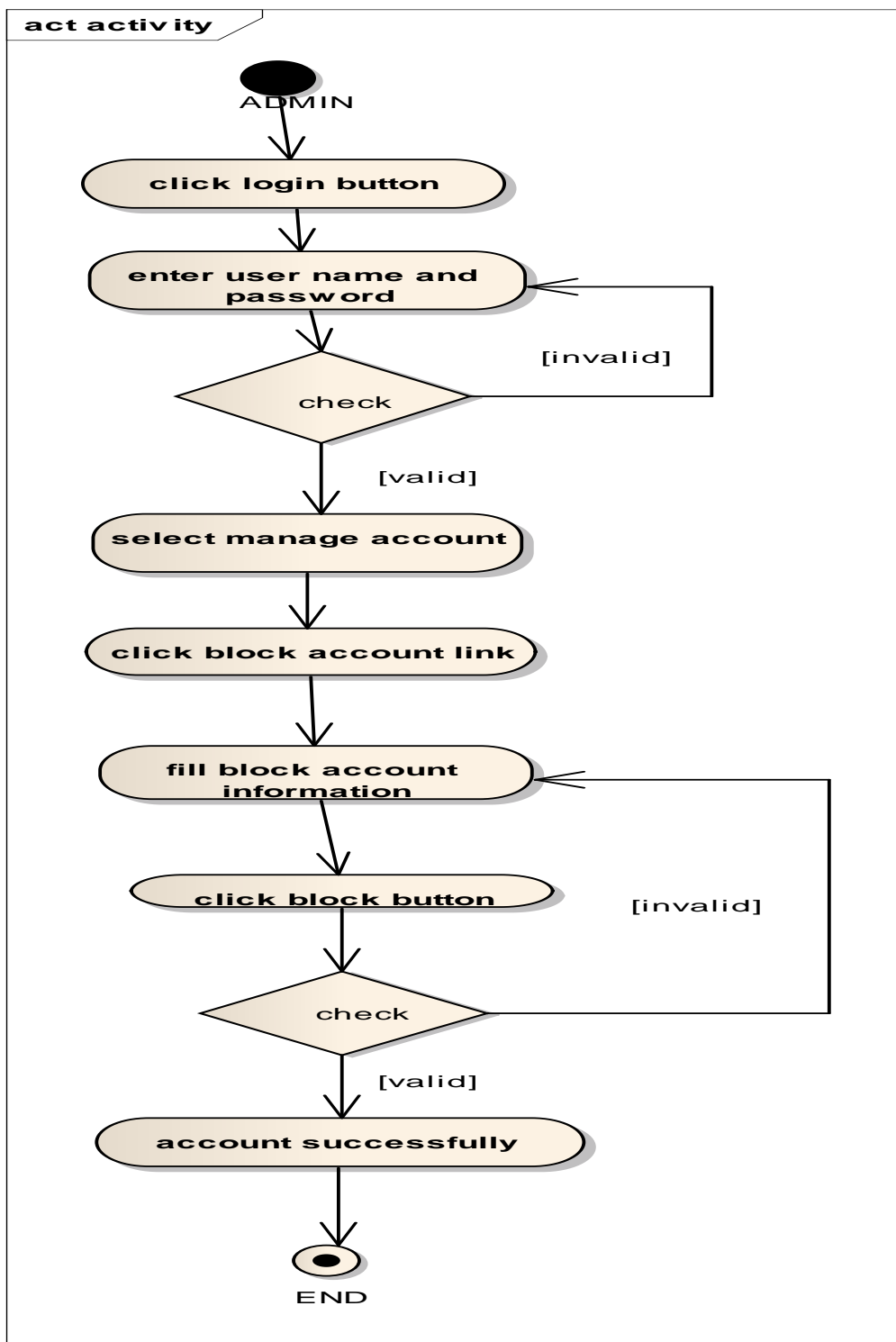


Figure 2.12: activity diagram for manage account

Chapter Three

System Design

3.1 Introduction

Design is the set of activities that needed to be performed by the designer. It deals with the transforming the customer requirements of SRS documents in to a form (set of documents) that is suitable for implementation in a programming language.

3.2 Goal of the Design

The goals of the design are: -

- Reduce the complexity for establishment of the new system.
- Show the best way to feasible output of the project.
- Minimize the extravagancy which is occurring due to done without design.
- Locate the necessary actor make easy and clear system development way.
- Avoid inappropriate thing which will can be the obstacle of the project.

Generally, design goals describe the qualities the system that the developers should consider:

- Security: - the system should authenticate its user by motivating them to enter the user name and password in order to get access to the system.
- Availability: - the system should be available every time that the user needs to access it.
- Usability: - the system should be having user friendly user interface to allow the user to interact with the system easily.
- Portability: - the system should be able to run on any mobile that supports android environment.
- Performance: - the system should give fast responses for user request. The main performance masher for the project is that time.
- Fault tolerance: - the system should be fault tolerant to where errors happen.
- Modifiability: - the system should be easily modifiable for further modification and enhancement of the application.
- Cost: -the system should be developed with minimum cost possible.

- End criteria: -the system should have simple and understandable graphical interface. All the interface, forms and buttons are written or designed in a simple language or common language so that they can access it without any difficult.

3.3 The Purpose of Design

- Is to describe how the new system is going to build and to obtain the information necessary for the device implementation of the system.
- This focused on understanding the model how the software will be built.
- It is the most input to indicate and show of the way to the implementation phase.

3.4 Class Modeling Diagram

Class diagram provide an over view of target system by describing the object and class inside the system and the relationship between them.it provide a wide variety of usages; from modeling the domain specific data structure to detailed design of the target system. With the share model facilities, you can reuse the class model in the interaction diagram for modeling the detailed design of the dynamic behavior. This diagram can be derived from one part of modeling is less responsibility collaboration (CRC). Below in the figure is shown the class diagram.

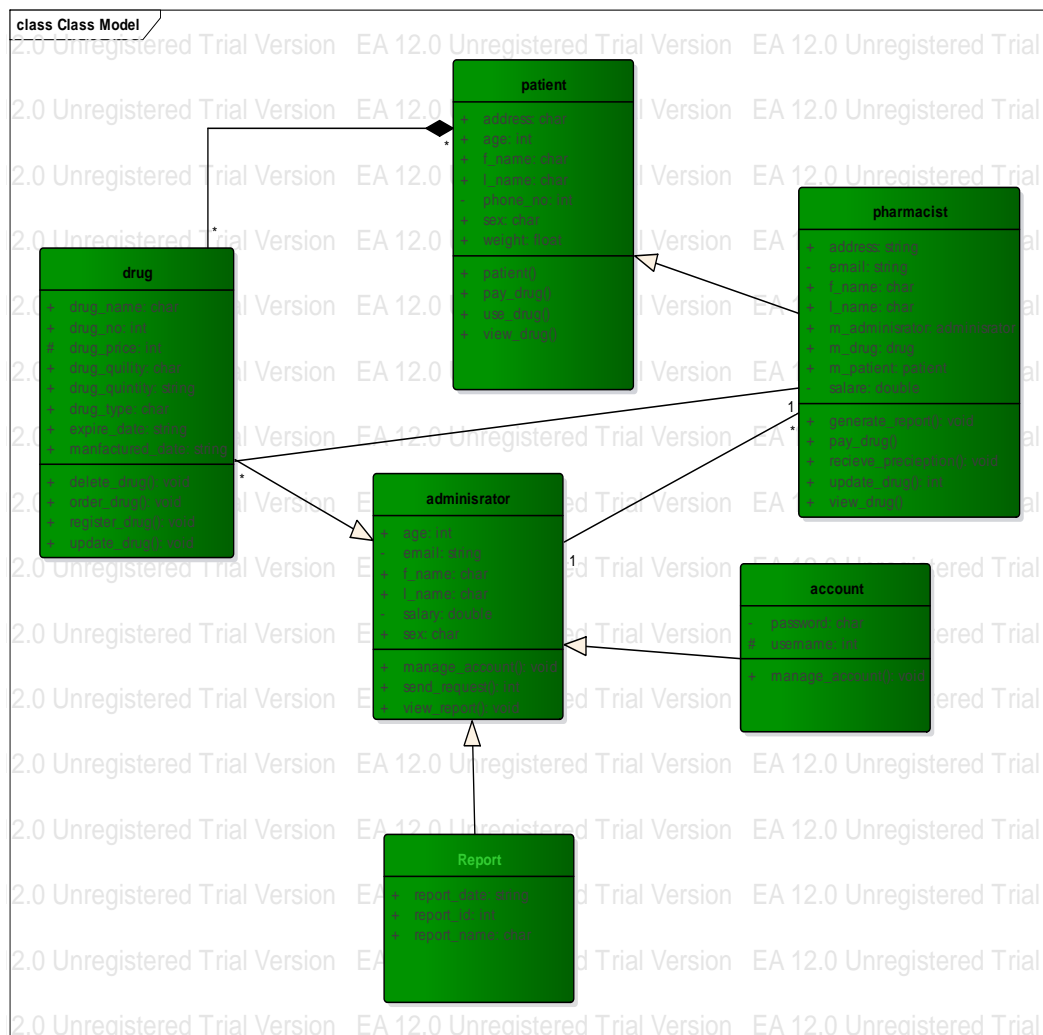


Figure 3.1: class diagram

3.5 Component Modeling

Component diagram shows how objects in our system will be grouped together and form components. The components interact with each other either in giving service to other components or requesting service from other component. Component diagram is partially useful with our system. Below is shown the figure for component diagram

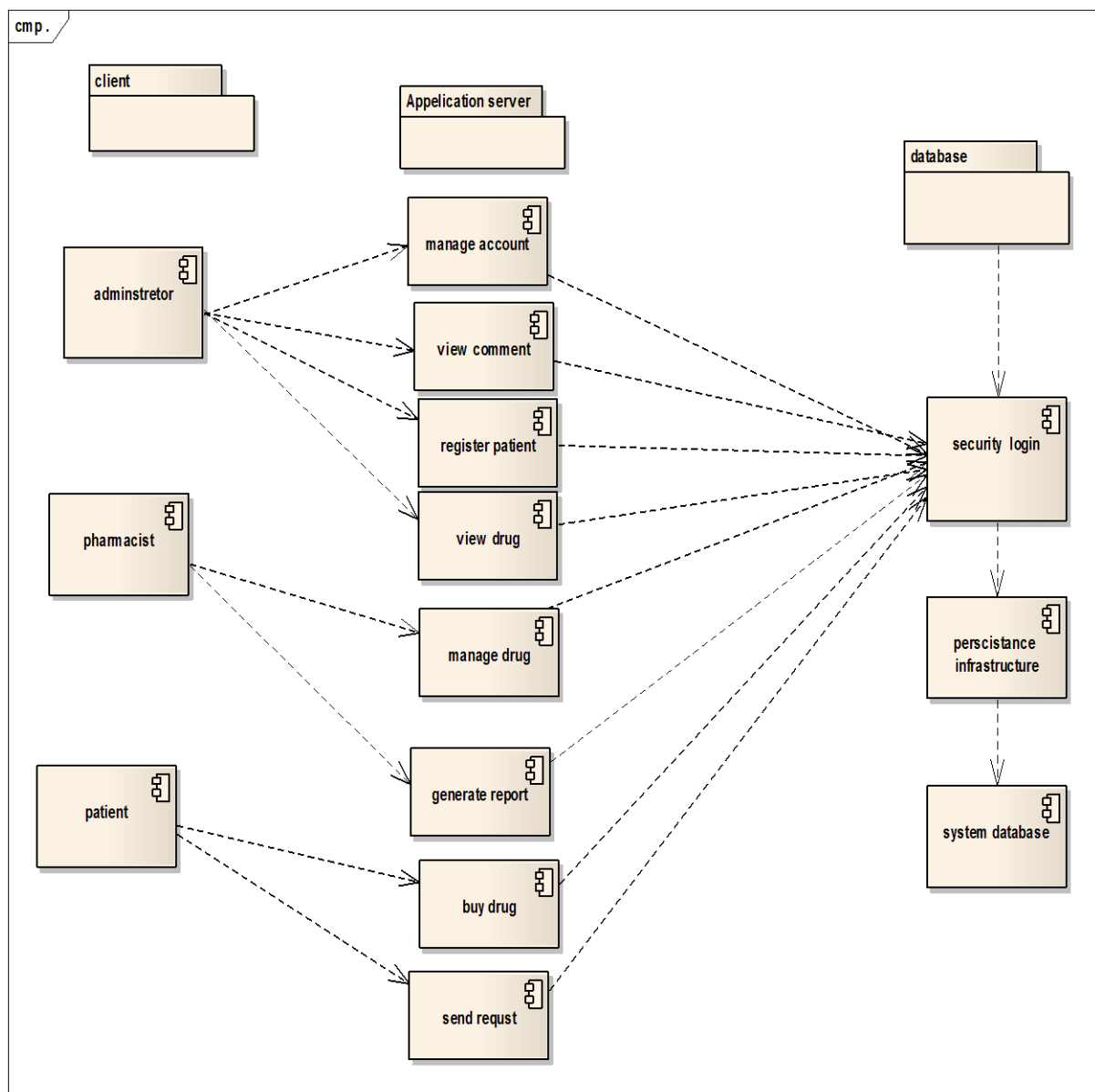


Figure 3.2: Component diagram

3.6 Deployment modeling

Deployment diagrams are used to describe the static deployment view of a system. In other words, deployment diagrams show the hardware of your system, the software installed on the hardware, and the middleware used to connect the disparate machines to one another. Deployment diagram shows how the system will be deployed on computer. In other words, it shows which component of the software will have installed on which machine and how they communicate with each other if they are on different machine. Deployment diagram can also create to explore the architecture of embedded system, showing how the hardware and software components work together.

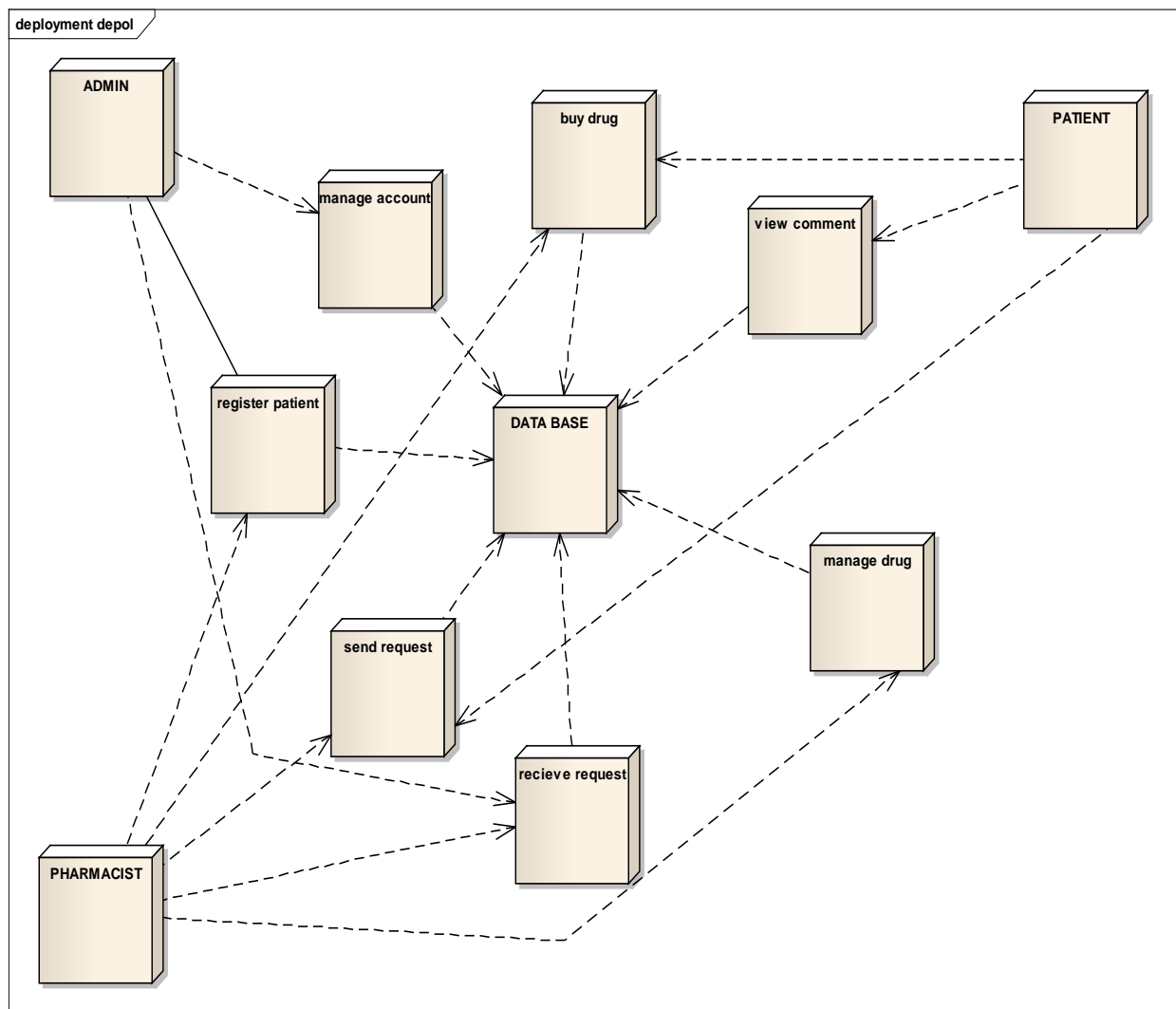


Figure 3.3: Deployment diagram

Chapter Four

Coding

4.1 Introduction

Implementation is a phase where objectives of physical operation of the system turned in to reality i.e. real working model. The crucial phase in the system development life cycle is the successful implementation of the new system design. The process of converting a new system in to an operational one is known as system implementation. This includes all those activities that take place to convert an old system to a new system. To implement the project, we use JAVA language.

4.2 Coding

First phase of implementation is coding. Coding is the process whereby the physical design specification created by the designers is turned into working computer code by the programmer can easily understand and work on that in future.

- We use JAVA programming language for the implementation of the system.
- Sample code is shown below:
- Sample code for viewing patient information

4.2.1 Forward engineering class diagram

- Forward (generate source code) engineering takes the UML class or interface model elements and creates a source code equivalent for future elaboration and compilation .
- By forward engineering code from the model the mundane work involved with having to key in classes and attributes and methods is avoided.

Sample code for forward engineering:-

Patient:-

```
Public class patient {
```

```
    Public String Address;
```

```
    Public int Age;
```

```
    Public char F_name;
```

```
Private int L_name;
Public int Phone_no;
Private char Sex;
Private double weight;
Public patient () {

}
Public void finalize () throws Throwable {

}
Public void Pay_drug () {

}
Public int use_drug () {
    return 0;
}
Public void view_drug () {
}
}
```

Drug extends Pharmacist:-

```
Public class Drug {
    Public char drug_name;
    Public int drug_no;
    Public double drug_price;
    Public int drug_quality;
    Public int drug_quantity;
    Public int expired_date;
    Public int manufactured_date;
```

```
Public Drug () {  
  
}  
Public void finalize () throws Throwable {  
  
}  
Public Void delete_drug () {  
    return null;  
}  
  
Public Void order_drug () {  
    return null;  
  
}  
Public Void register () {  
    return null;  
}  
Public Void update_drug () {  
    return null;  
}  
Public class Pharmacist extends Drug {  
    Public String address;  
    Public char administrator;  
    Private char email;  
    Public int f_name;  
    Public int L_name;  
    Private int Phone;  
    Public float salarie;
```

```
Public int sex;

Public Pharmacist () {

}

Public void finalize () throws Throwable {

}

Public Void generate_report () {
    return null;
}

Public Void pay_drug () {
    return null;
}

Public Void recieve_prescription () {
    return null;
}

    Public Void view_drug () {
        return null;
    }

}}
```


4.2.2 Reverse engineering

Reverse engineering in enterprise architect (EA) enables to import existing source code from a variety of code languages into a UML model.

Reverse engineering enables users to examine legacy code and examine the functionality of code libraries for reuse or to bring to date with the code that has been developed as a part of process called synchronization.

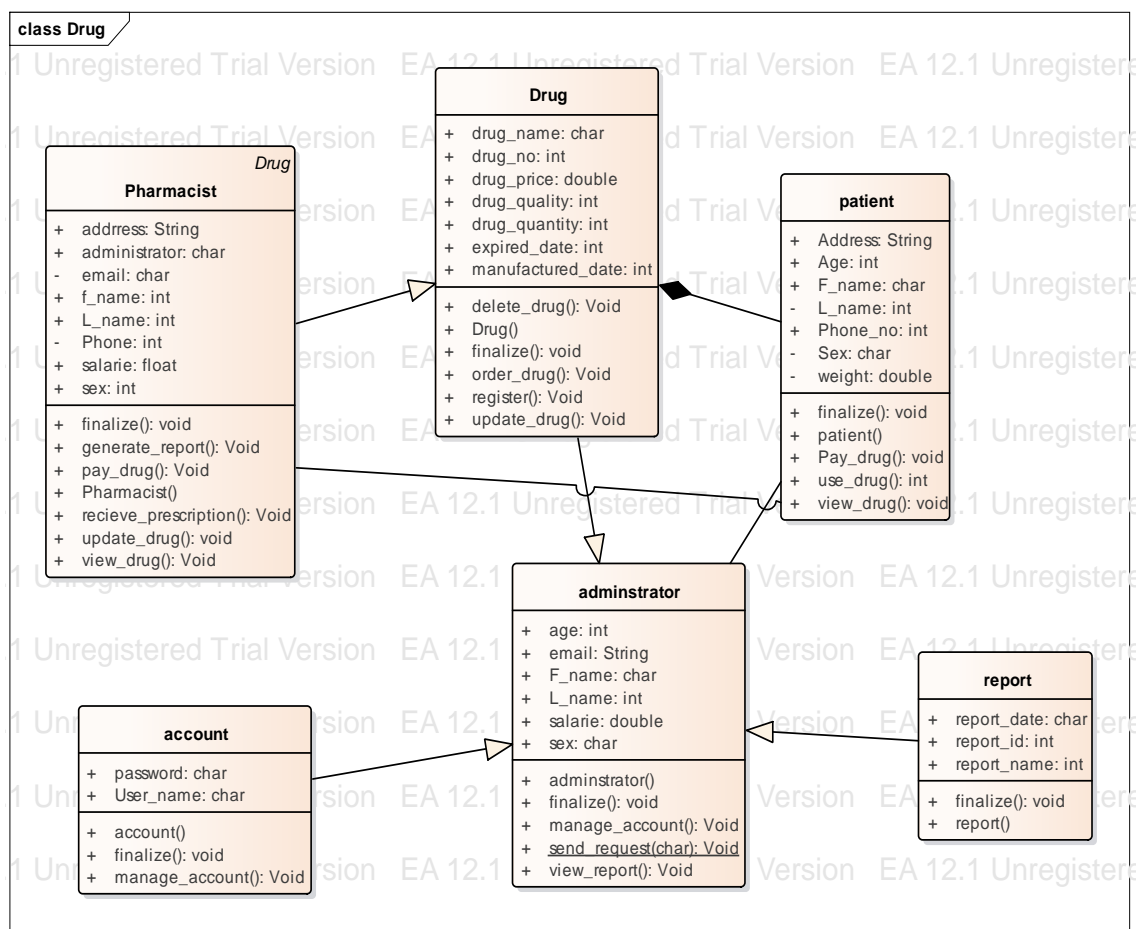


Figure 4.1 reverse engineering diagram

4.3 Testing

Testing is a final phase of implementation. Testing is a process to show the correctness of the program. Testing is checking of the system workability in an attempt to discover errors and avoiding such errors from the system. Some examples of testing.

4.3.1 Unit Testing

Each module is tested individually in an attempt to discover any errors in its code. In unit testing each module (roughly a section of code that performs a single function) is tested individually to discover any errors that may exist in the modules code.

| Unit test :Authentication for login | | |
|---|-------------------------------------|---|
| Assumptions = Login into appropriate page | | |
| Test Data = User Name(empty , invalid user name, valid username) and Password (empty ,invalid password, valid password) | | |
| Steps to be Executed | Data | Expected results |
| Empty Username and password to Click Login button | ----- | "Please enter field set!" |
| Enter invalid Username, password and Click Login button | Any valid data for the other fields | "Invalid Username or Password. Please Enter Again!" |
| Enter valid user name, empty password and Click Login button | Any valid data for the other fields | "Please Enter feildset" |
| Enter valid password ,empty user name and Click Login button | Any valid data for the other fields | "Please Enter field set" |
| Enter valid Username and Password then Click Login button | All fields Fulfill with valid data | "Login in to appropriate page" |

Table 4.1 unit testing

Chapter Five

Conclusions and Recommendation

5.1 Conclusions

Due to the abundance of patients and the growing number of operations at the pharmacy, the handling of paper work and the updating of records had been difficult. The purpose of this project was to enable the pharmacy to carefully document the entire necessary patient and the related information and to manage and manipulate patient's information effectively. The system also helps them for the proper functioning of their operation in the considerable time and accuracy. With all of that in mind, the newly proposed system will allow pharmacy person including system administrators and pharmacist to maintain and manage patient's information and transactions more efficiently and securely. The system is flexible, accurate and attractive with easy GUI approach. Generally, the team confidently ensures that the software is completed successfully with negligible errors. Finally, the team also expects that the developed system will change the general drug management of Kombolcha pharmacy drug store and makes it more reliable and efficient than the previous manual system.

5.2 Recommendation

The system we have developed is an application computerized system it needs a skilled person to work with the system. So, we recommend the system should be required the responsible and skilled person. We highly recommend the system should be kept in highly safe and favorable condition and we also recommend that to enhance the performance and functionality of the system the next developer can include the following tasks:

- The system includes financial cases.
- Back up will automatically generated every 24 hours.
- The system will cover all Ethiopian customers.
- Anyone can fill our system's limitation due to we were faced within shortage of time to more focus on this project.

Références

☞ www.google.com

☞ Software engineering tools and practice, 2nd edition.

☞ Our teacher as an advisor

☞ Students of Software Engineering, 2nd year