ASHVIN

WORKFLOW MONITORING SYSTEM

Software Design Specification

Version 1.3

Adviser: Sunny Wong

Stakeholder: Siemens Healthcare
Robert Neff

Group Members: Brian Courtney

Dan Huynh

Daniel Molander

Michael McLaughlin

Sunny Patel

Revision History

Name(s)	Date	Changes	Version Number
Sunny Patel	2/19/2014	Added Data Layer, Data Layer components, Database Design	1.4
Dan Molander	2/17/2014	Added presentation layer components, summary, presentation layer technologies used, and design features	1.3
Brian Courtney Dan Huynh Dan Molander Mike McLaughlin Sunny Patel	2/16/2014	Updated Table of Contents	1.2
Dan Huynh Dan Molander	2/14/2014	Added Table of Contents	1.1
Mike McLaughlin	2/12/2014	Initial document creation	1.0

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose

This document specifies the hardware and software architecture of the Ashvin Workflow Monitoring System. The designs described herein are the implementations of the requirements defined in the Ashvin Requirements Document.

This document concludes with a summary of the perceived advantages and disadvantages of the design included herein, as assessed by the developers of the Ashvin Workflow Monitoring System.

1.2 Scope

This document describes the design of Ashvin Workflow Monitoring System v1.7. This document covers the design details necessary to implement the Must Have priorities of Ashvin, as described in the Ashvin Requirements Document. This document does not cover the design details necessary to implement Should Have or Must Have priorities.

The intended audience for this document consists of the Ashvin Workflow Monitoring System developers, stakeholders and system testers.

1.3 Definition of Terms

Active Scanning System – Electronic system where the devices present in a patient care setting autonomously scan their environments for input without the need for a human to initiate the scanning processes. E.g., a computer program parses live camera images for barcodes and scans them into the computer on its own. See also: **Passive Scanning System**.

Android – Mobile operating system (OS) maintained by Google. See also: **Mobile Device**, **Smart Device**.

Barcode – Machine readable label placed on an object used to import data into a computer. Barcodes can be read optically via a LASER scanner or a camera.

Business Layer – Layer of system architecture that is responsible for the logic processing. This may include camera image processing, data base access or other tasks necessary for the **Presentation Layer** and **Data Layer** to interact.

Clinician – Person, such as a nurse or doctor, responsible for patient care on a daily basis.

Data Layer – Layer of system architecture that is responsible for the patient data. For Ashvin, this includes the **EHR**.

Electronic Health Records (EHR) System – Hospital storage system for patient information. In regards to this document, the term EHR encompasses the entire hospital records system, including, but not limited to, the database information for a given patient, along with other hardware / software that is used in the patient care process.

Five Rights of Medication Administration – Medical industry standard for ensuring the accurate administration of medication to a patient. The rights are:

- *Right Patient*
- *Right Medication*
- *Right Dose*
- *Right Time*
- *Right Route of Administration*

First normal form - Very basic rules for an organized database:

- Eliminate duplicative columns from the same table.
- Create separate tables for each group of related data and identify each row with a unique column (the primary key).

General Route – A medication general route refers to the manner of administration for the medication. An example of an incorrect general route would be if a patient has to receive a medication in pill form as per the medication's instructions but receives the medication in liquid form.

GlassFish – Java-based open source application server that provides portability and scalability to the Ashvin server using the REST technology.

Google Glass – Interactive glasses worn by a person that allows him/her to digitize the world around him/her through the use of an attached camera. The glasses also display information onto the glasses for the wearer to view as if looking at a computer monitor.

Graphical User Interface (GUI) – Interface used by a user to graphically interact with a hardware or software system.

Image Recognition – Technology involving optical recognition by a computer of a real world object.

Information Technology (IT) – Adjective used to describe any component of the Ashvin Workflow Monitoring System that is computer related.

Infrared (IR) – Wavelength of light just outside of the visible range.

Intravenous (IV) Medication Administration – Route of administering medication to a patient via the vein using a needle.

Mobile Device – A digital device that is contained on the person of a user and is used to receive output from the EHR or video source, as well as to input information into the EHR.

Passive Scanning System – Electronic system where the devices present in a patient care setting scan their environments for input with the help of a human being. E.g., a person explicitly clicks a button on a hand held scanner to read a barcode. See also: **Active Scanning System**.

Prescription – Formal, physician implemented medication plan directly related to a given patient.

Presentation Layer – Layer of system architecture that is responsible for the client-based **GUI**. Provides **Clinician** or **System Administrator** access to the system.

Radio Frequency Identification (RFID) – Wireless system for passing information between objects using an **RF**.

Real-time Locating Systems (RTLS) – System used to identify and track objects in a live environment. An RTLS may be implemented via **RFID**, **IR** or other form of technology.

Representational State Transfer (REST) - is an architectural style consisting of a coordinated set of constraints applied to components, connectors, and data elements, within a distributed hypermedia system.

RF – Radio Frequency.

Route - the manner of administration for a medication

Second Normal Form (2NF) - After completing first normal form, the database must satisfy these rules also:

- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
- Create relationships between these new tables and their predecessors through the use of foreign keys.

Smart Device – See **Mobile Device**.

System Administrator – IT person responsible for basic setup and upkeep of the Ashvin Workflow Monitoring System.

Third Normal Form (3NF) - After database comply with 1NF and 2NF, it must also follow this rule to be considered in 3NF.

- Remove columns that are not fully dependent upon the primary key.

User – Any person, typically a **Clinician** or **System Administrator**, who uses the Ashvin system.

1.4 Context Diagram

Shows the Ashvin Workflow Monitoring System context diagram.

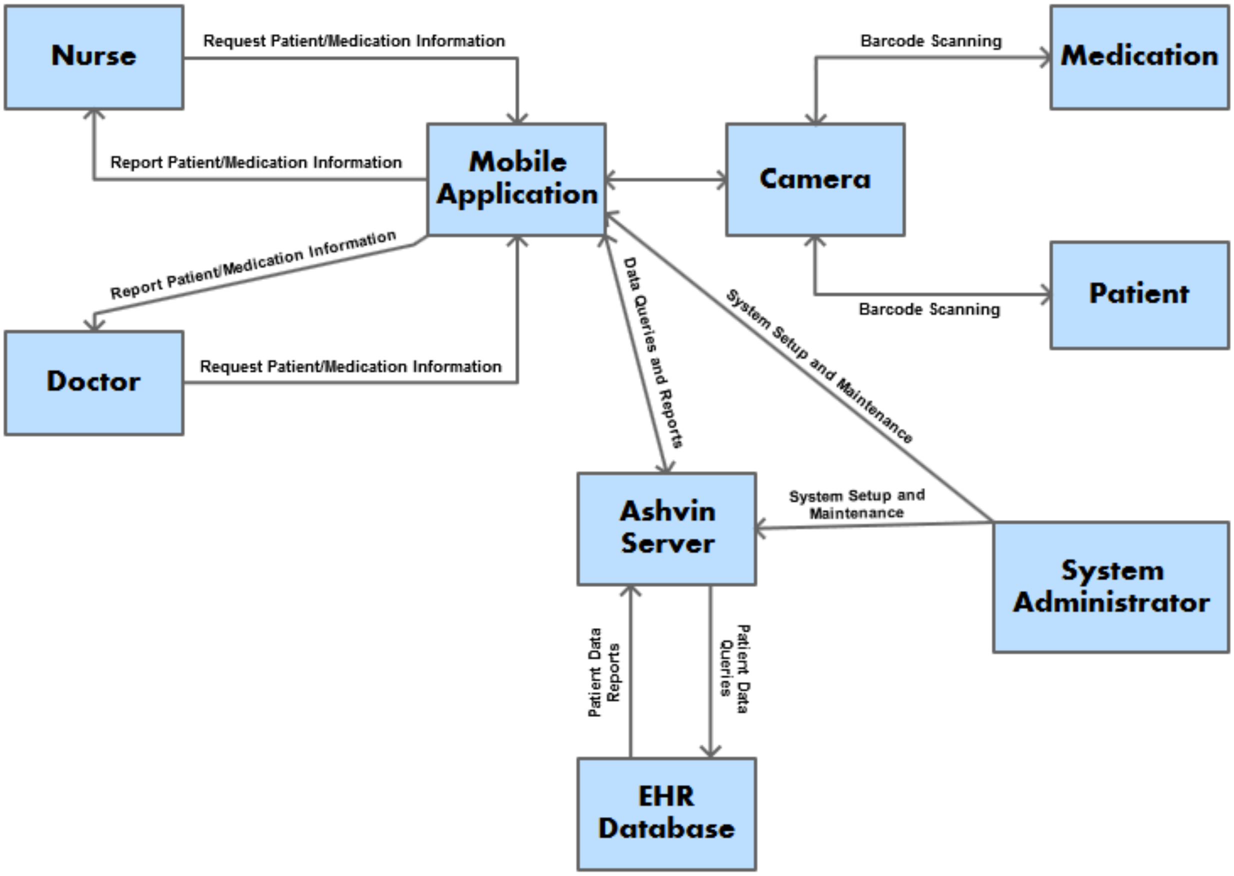
Nurse
Request Patient/Medication Information
Report Patient/Medication Information
Mobile Application
Camera
Barcode Scanning
Medication
Patient
Barcode Scanning
Report Patient/Medication Information
Doctor
Request Patient/Medication Information
Data Queries and Reports
System Setup and Maintenance
Ashvin Server
System Setup and Maintenance
System Administrator
Patient Data Reports
Patient Data Queries
EHR Database

Figure 1 Ashvin Workflow Monitoring System context diagram

2 Architecture

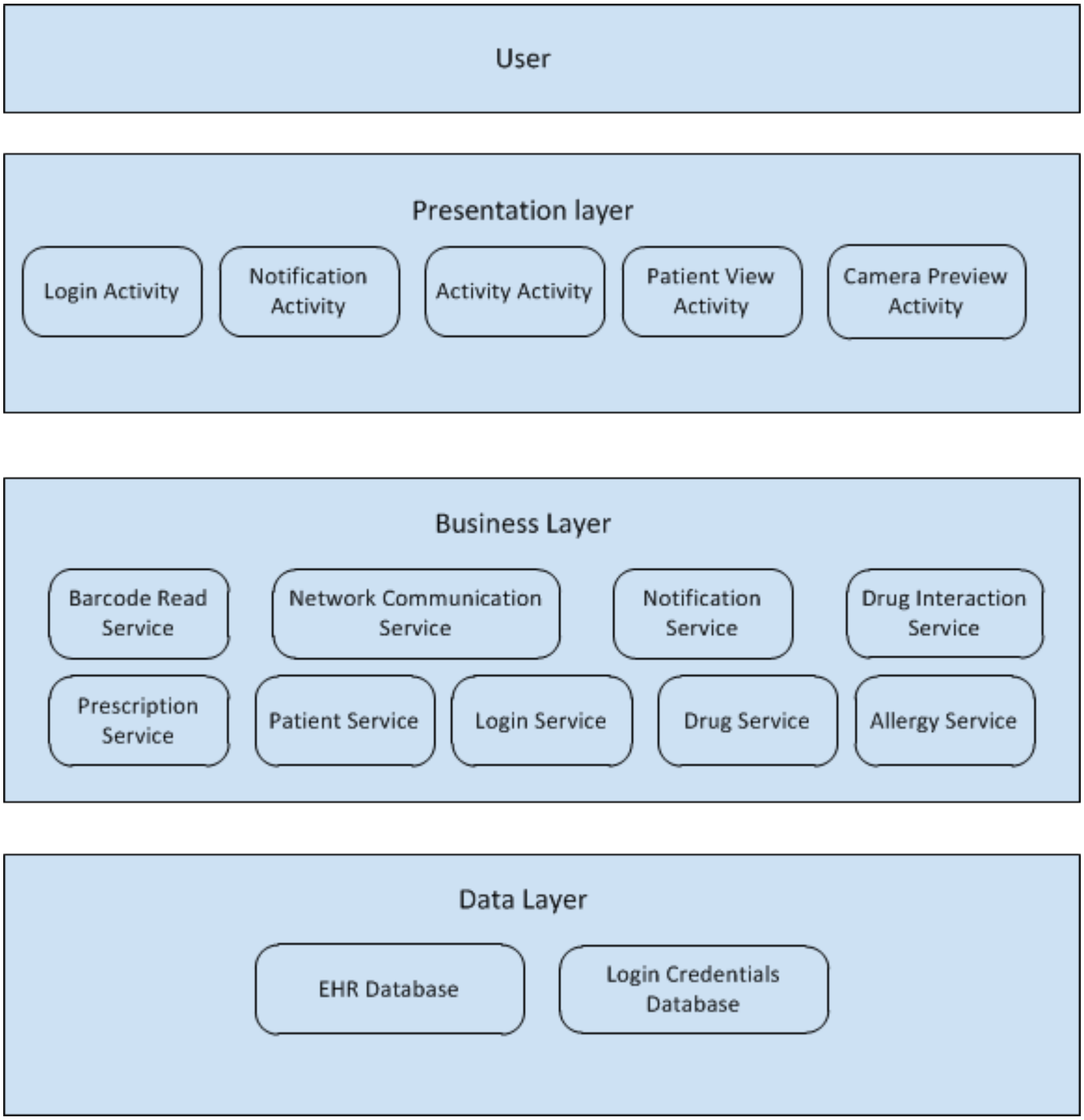

Figure 2 Architecture Diagram

2.1 Overview

Ashvin follows a three-tier design, separating the data, business, and presentation components. This keeps the application simple and modular, so that it can be easily expanded and upgraded.

2.2 Three-Layer architecture

The Ashvin system conforms to the Three-layer architecture, which includes the presentation layer, the logic layer, and the data layer [Figure 2]. The separation of the three layers provides developers and system administrators the ability to cultivate each layer independently.

Since Ashvin system is an integration of different departments in the healthcare system, the Three-layer architecture is the best solution to develop a portable, maintainable, and high performance system.

2.3 Survey of Technology Used

2.3.1 Presentation Layer

All Presentation Layer technologies components are developed using the existing Android SDK. The Android SDK provides access to interactive graphical widgets that can be arranged dynamically inside a layout; this allows for fast development of user interface views, using highly tested, well documented widgets.

2.3.2 Business Layer

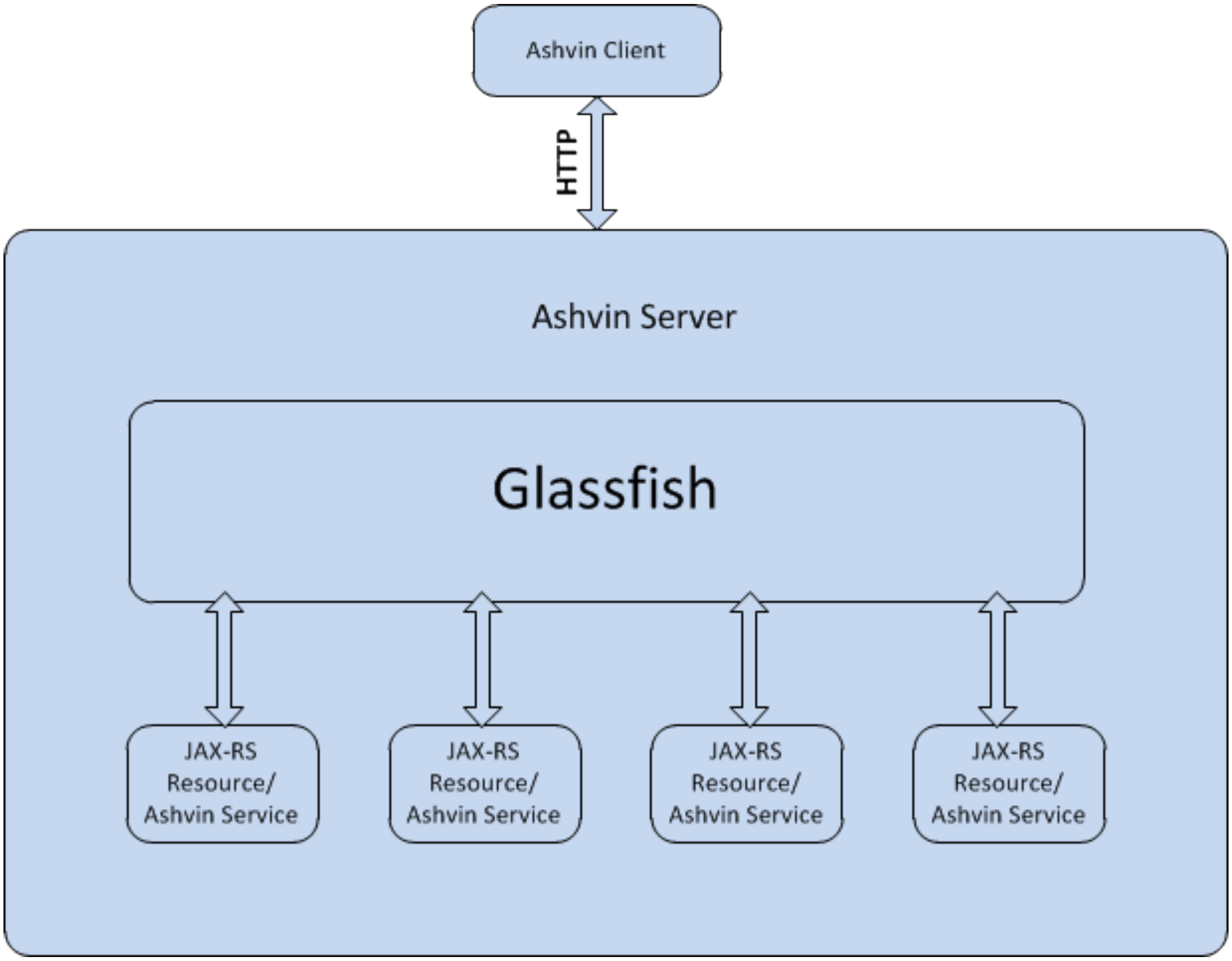
Ashvin Client
HTTP
Ashvin Server
Glassfish
JAX-RS Resource/ Ashvin Service
JAX-RS Resource/ Ashvin Service
JAX-RS Resource/ Ashvin Service
JAX-RS Resource/ Ashvin Service

Figure 3 Business Layer server

The technology behind the Ashvin server is the Glassfish 4.0 application server, JAX-RS 2.0, and Java EE7. Glassfish's primary feature is to handle the HTTP requests from the Ashvin clients and map them to the corresponding JAX-RS resources [Figure 3]. It is known to be portable and high scalability.

JAX-RS is a Java RESTFul web service API. It's designed to work cooperatively with the Glassfish application server. Java application implemented with JAX-RS can be hosted on Glassfish as JAX-RS resources. The major advantage of JAX-RS API is annotations, which helps simplify the development and deployment of web services.

The majority of the business layer components reside on the server; however, the mobile client is responsible for detecting and decoding barcodes through the use of the *zBar* library. No analysis takes place here and the data is immediately transmitted to the server.

2.3.3 Data Layer

The Ashvin Data layer stores all the information required to perform all the application's functionality. This includes patient information, drug information, and user login information. In order to perform the tasks of Ashvin's users our database needed to fulfill the non-functional requirements listed here:

- Scalability
- Cost
- Robustness
- Flexibility of architectures
- Learning curve for team

With these non-functional requirements laid out, we then proceeded to research relational databases as well as NoSql databases. The following is a list of databases we considered.

- Oracle 12c
- IBM DB2
- SQLite
- Mongo DB (NoSql)
- MySQL

After pulling our data together, we quickly realized NoSql databases will not be able to represent our data effectively. Next consideration was SQLite, because it is self-contained, very easily portable and no configuration this database was a good candidate for the flexibility of architectures requirement. However SQLite lacked in performance and fine-tuning functionality that the remaining databases provided. Oracle 12c and IBM DB2 were simply too costly and learning curve for proper configuration/implementation would be very time consuming. MySQL was the our final choice because it provided the simple functionality we needed, pretty simple configuration, very robust, and the relational SQL database would be able to represent our data much better than Mongo DB.

2.4 Presentation Layer Components

2.4.1 Mobile Application

2.4.1.1 Login Activity

Requirements Satisfied: FR-022

The Ashvin login activity is responsible for receiving the username and password from the user, validating, and launching the main activity. The login activity

2.4.1.2 Activity Activity

Requirements Satisfied: FR-005, FR-012, FR-013, FR-014, FR-015, FR-016, FR-017, FR-018, FR-019, FR-020

The activity activity is responsible for displaying scanned medications. The activity activity receives medication information from the network communication service.

2.4.1.3 Patient View Activity

Requirements Satisfied: FR-004

The patient view activity is responsible for displaying scanned patient information. The patient view activity receives patient information from the network communication service

2.4.1.4 Camera Preview Activity

Requirements Satisfied: FR-003

The camera preview activity is responsible for displaying a preview of the device camera. The camera preview activity interacts directly with the Android SDK.

2.4.1.5 Notification Activity

Requirements satisfied: FR-012, FR-013, FR-014, FR-015, FR-016, FR-017, FR-018, FR-019, FR-020

The notification activity is responsible for notifying the user that an alert has been detected; this includes patient allergies, medication route warnings, as well as medication/patient scanned events.

2.4.2 Google Glass Application

A client application running on Google Glass provides the nurse with an unobtrusive, intuitive interface to the system. The front camera on the device passively scans the nurse's field of vision for barcodes. When one is detected, it transmits the code to the server for analysis. The app presents messages from the server to the nurse via the head-mounted display in the corner of his or her field of vision. This includes prescription administration information, patient data, and contextual warnings if the system believes that the nurse is about to make a mistake.

The Google Glass application is a variant of the standard mobile client, and it is the primary intended method of using the system. Unlike running the client app on a phone or tablet device. Glass allows for passive scanning. It does not require the user to manually scan a code.

2.5 Business Layer Components

2.5.1 Mobile Application

2.5.1.1 Network Communication Service

The network communication service provides network connection between the Ashvin client and the Ashvin server to exchange data.

2.5.1.2 Barcode Read Service

Requirements Satisfied: FR-004, FR-007, FR-008, FR-009

The barcode reading service analyzes the video stream from the client cameras to detect and read barcodes.

2.5.1.3 Notification Service

The Ashvin notification service is responsible for activating the notification activity. the notification service interacts with the network communication service, and receives activates the notification activity upon receipt of data from the network communication service.

2.5.2 Server Application

2.5.2.1 Login Service

Requirements Satisfied: FR-006, FR-022

The login service authenticates the Ashvin client users to provide or prevent access to the Ashvin system based on the credentials provided and the user type.

2.5.2.2 Drug Service

Requirements Satisfied: FR--014

The drug service maps the drug object to the EHR database to provide the drug information to the Ashvin client.

2.5.2.3 Patient Service

Requirements Satisfied: FR--012

The patient service provides Ashvin client access and modification to the patients' data.

2.5.2.4 Drug Interaction Service

The drug interaction service provides a list of the drugs that might have adverse affects with the drugs that the patient is currently taking.

2.5.2.5 Allergy Service

The allergy service provides a list of foods and drugs that the patient is allergic to.

2.5.2.6 Prescription Service

Requirements Satisfied: FR-011, FR--014

The allergy service provides access and modification to the patient's prescription on the EHR database.

2.5.2.7 Barcode search Service

The barcode search service provides the available data that associated with the requested barcode.

2.6 Data Layer Components

2.6.1 EHR Database

Requirements Satisfied: FR-002

The Ashvin Workflow Monitoring System EHR database serves as a mock up for a real EHR. This database is constructed as a tool for the developers to use in developing the necessary logic for object recognition. Subsequent releases of the Ashvin system will integrate the ability to interface with a real EHR.

The table structures of the mock EHR database are found in this section.

2.6.2 Login credentials Database

Requirements Satisfied: FR-022

Login credentials database stores all the information required to authenticate Ashvin users. This database was separated from our mock EHR database because we wanted the ability to swap EHR databases future releases of Ashvin and not have it interfere with the authentication process of our users.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
user_id	int(11)	-	No	None	AUTO_INCREMENT	PRI
username	varchar(25)	-	No	None	-	
password	varchar(32)	-	No	None	-	MUL

Table 1 : Users Table

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
user_id	int(11)	-	No	None	AUTO_INCREMENT	PRI
first_name	varchar(25)	-	No	None	-	
middle_name	varchar(15)	-	Yes	None	-	
last_name	varchar(25)	-	No	None	-	

Table 2 Users_information Table

We normalize our database to third normal form, however there maybe some rules violated because we used our best judgment.

3 Design Features

3.1 Login

The Login feature allows a user to enter their username and password, and allow them into the application upon successful verification, or deny access upon unsuccessful verification.

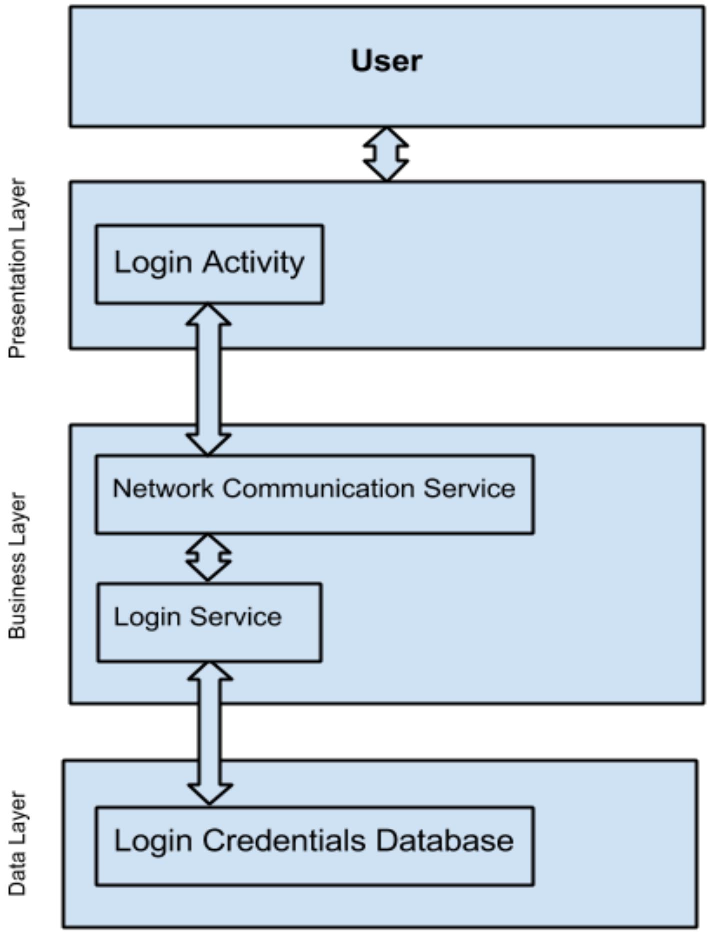

Figure 4 Login Process

The login process starts by accepting the username and password from the user and passing it to the network communication service. The network communication service then calls over the network to the login service, which reads from the login credential database where the username and password will be checked for validity. After checking validity, the success or failure status is passed from the network communication service to the login service, and then back to the login activity. If the login activity receives a successful verification it will launch the main application, if the login fails, the login activity will display a warning to the user that the username or password was not correct.

3.2 Risk Notification Feature

The risk notification feature allows a user to be notified when medication, and patients are scanned, as well as when potential dangers are determined to be present.

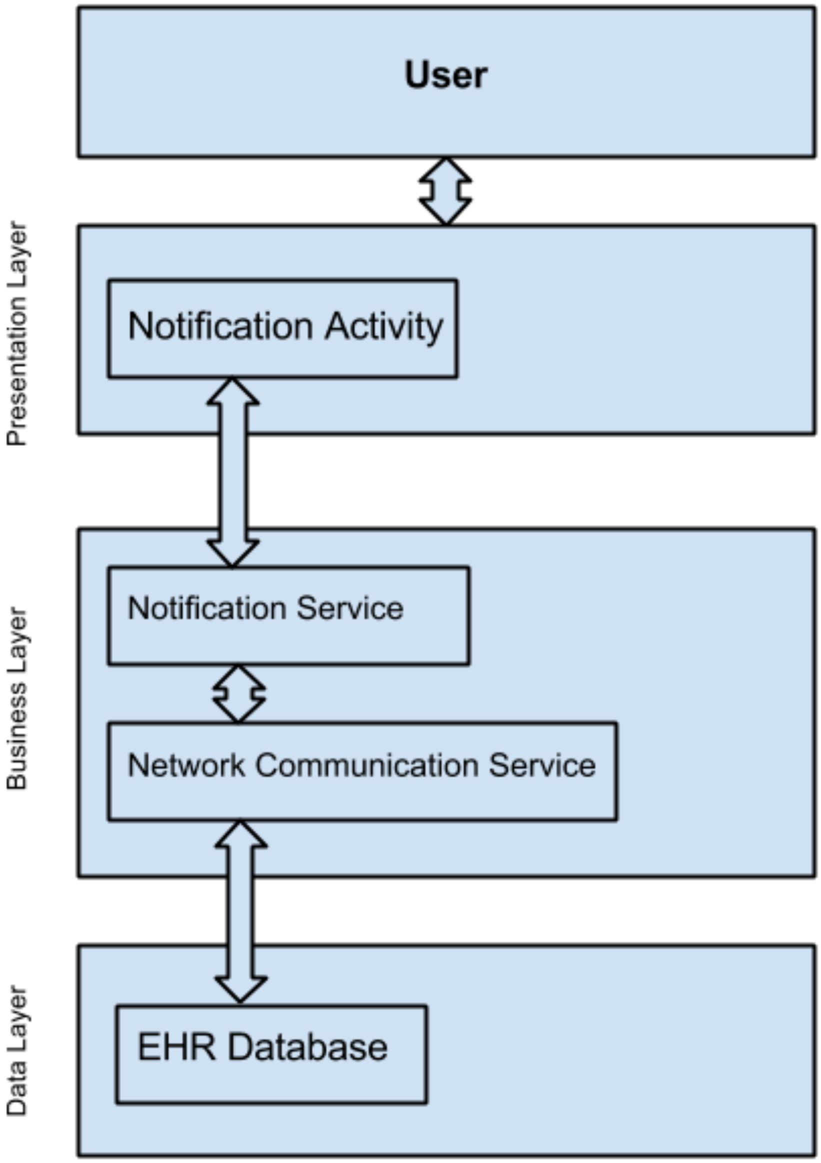

Figure 5 Risk Notification Process

The risk notification process starts with the after the completion of a barcode scan, or other event. After the data from the data is passed from the EHR database to through the network communication service to the mobile application, the mobile will notify the notification service in addition to other services needed for the event. The notification service then activation the notification activity for renderings to the screen.

3.3 Patient Information Feature

The patient information feature allows a user to view the information stored in the database for the last scanned patient.

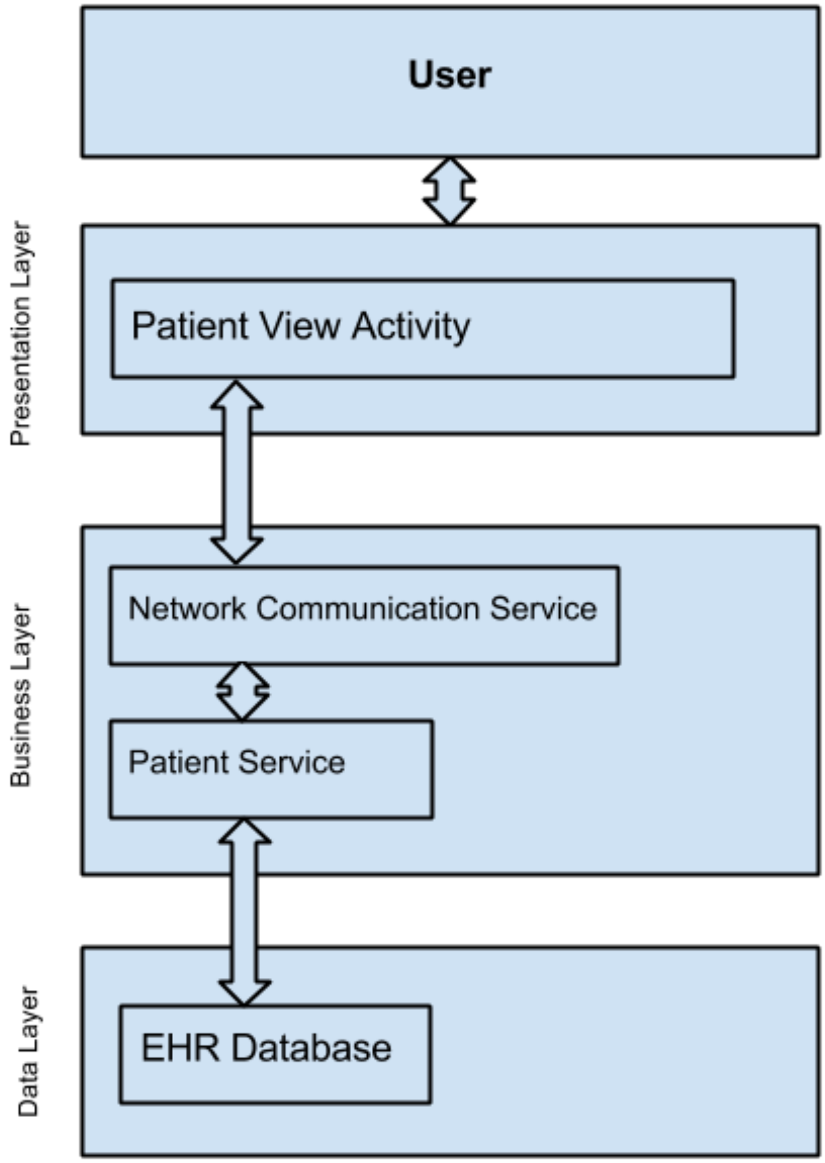

Figure 6 Patient Information failure

The patient view population process starts when a patient barcode is scanned, and the returning data is passed to the patient service on the server. The patient service then passes the necessary data through the network communication service to the patient view activity for rendering to the screen.

3.4 Patient Prescription Feature

The patient prescription feature allows a user to view the current prescriptions associated with the last scanned patient. This feature lists all of the medications, dosages, and route associated with each prescription.

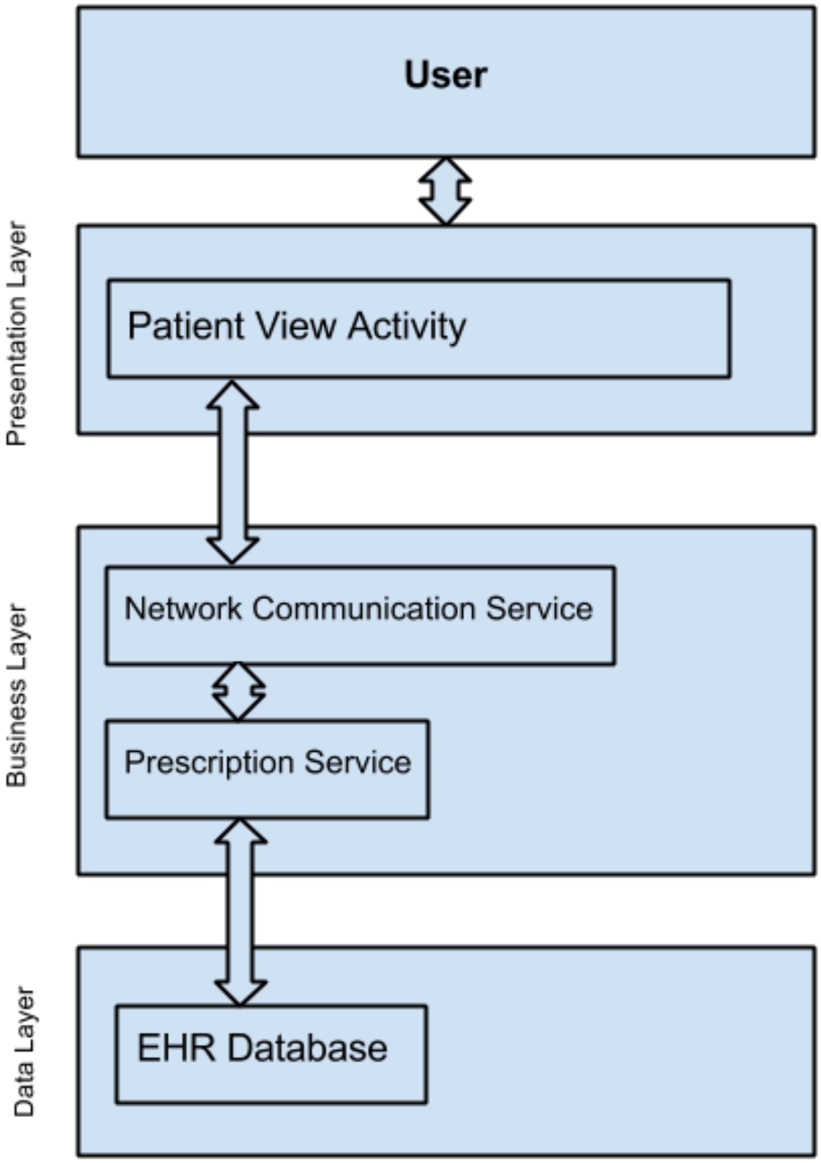

Figure 7 Patient Prescription service

The patient prescription population process starts when a patient barcode is scanned, and the returning data is passed to the prescription service on the server. The prescription service then passes the necessary data through the network communication service to the patient view activity for rendering to the screen.

3.5 Active Barcode Scan Feature

3.5.1 Scanning for Patients

The scanning for patients feature allows a user to scan a barcode, and pull patient data from the EHR database. This feature is fundamental to patient view feature, as it provides the patient id.

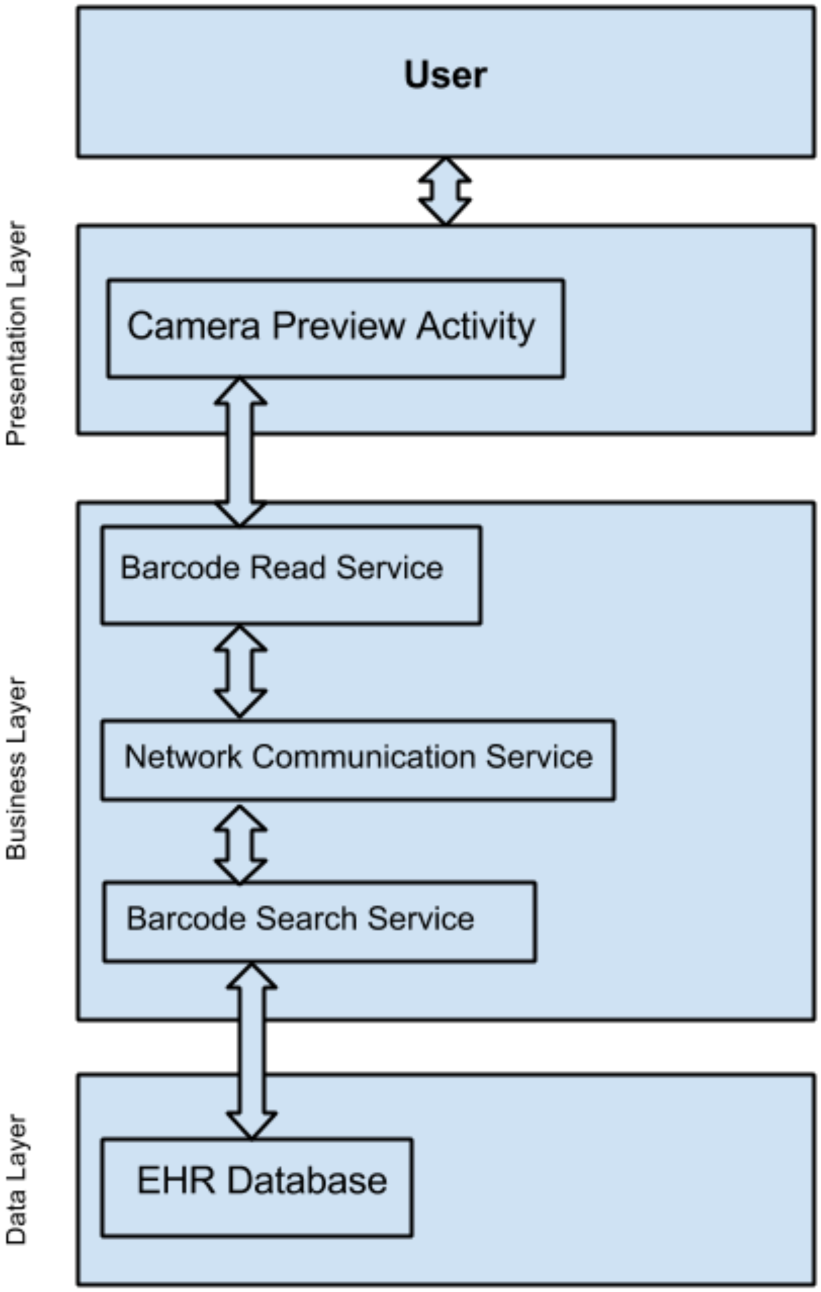

Figure 8 Scanning for Patients process

The patient scan process starts when a patient barcode passes into the field of vision of the mobile device’s camera. The image is converted to a string by the barcode read service, which is then passed through the network communication layer to the barcode search service. The barcode search service then looks up the relevant data in the EHR database, and passes it back through the network communication service to the barcode read service to notify the client that the request was successfully received.

3.5.2 Scanning for Medication

The scanning for medications feature allows a user to scan a barcode, and pull medication data from the EHR database. This feature is fundamental to medication view feature, as it provides the medication id.

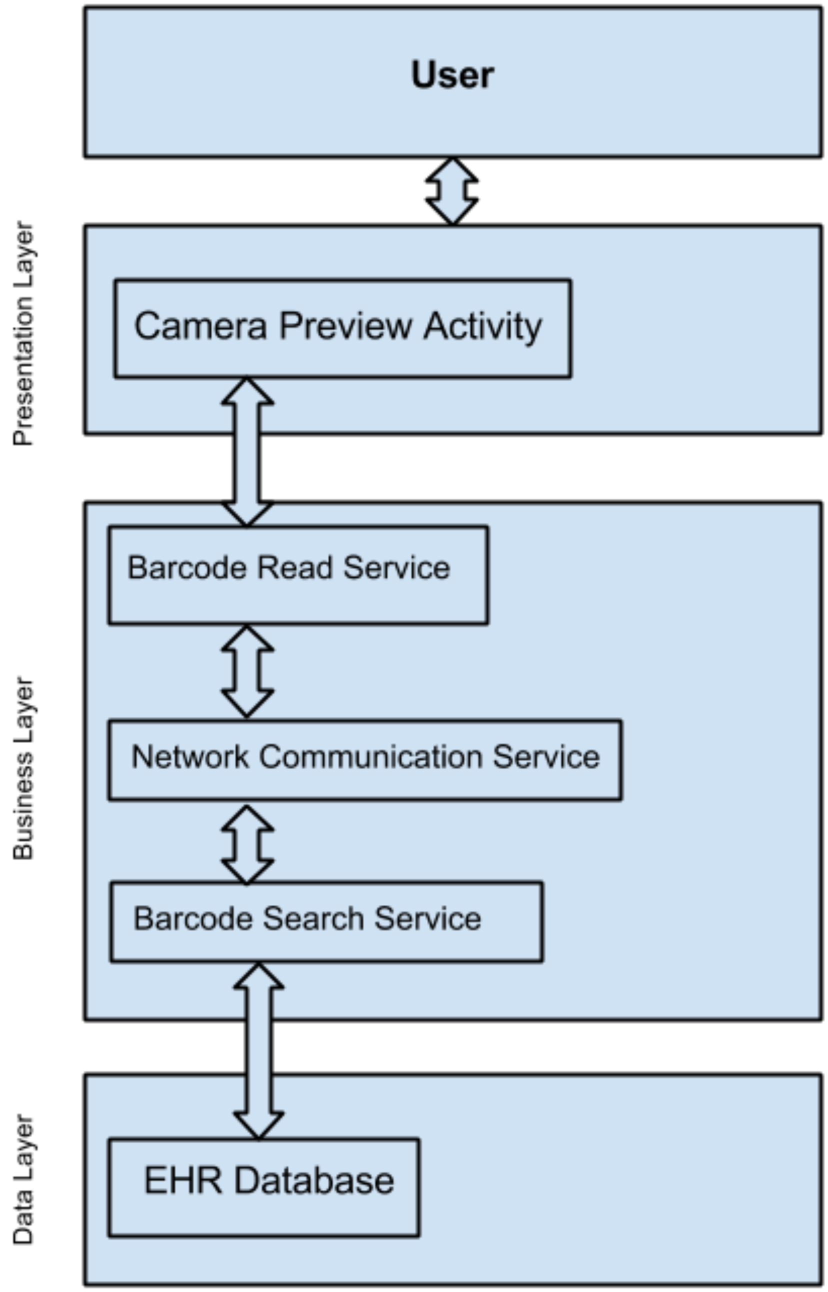

Figure 9 Scanning for Medication Process

The medication scan process starts when a medication barcode passes into the field of vision of the mobile device's camera. The image is converted to a string by the barcode read service, which is then passed through the network communication layer to the barcode search service. The barcode search service then looks up the relevant data in the EHR database, and passes it back through the network communication service to the barcode read service to notify the client that the request was successfully received.

4 Database Design

In order to effectively performance tune our database we wanted to ensure we organized our database efficiently. Following the principles of database normalization we attempted to organize our data tables to third normal form. Third normal form requires us to ensure we have met the requirements of first normal form and second normal form. This will help when performance tuning our mock EHR database as we add more patients and medications and in general scaling Ashvin.

4.1 Address Table

This table stores generic addresses that can be associated with people or places.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
address_id	bigint(20)	UNSIGNED	No	None	AUTO_INCREMENT	PRI
address1	varchar(30)	-	Yes	NULL	-	-
address2	varchar(25)	-	Yes	NULL	-	-
apt_number	varchar(10)	-	Yes	NULL	-	-
city	varchar(20)	-	Yes	NULL	-	-
state	varchar(20)	-	Yes	NULL	-	-
zip_code	varchar(10)	-	Yes	NULL	-	-

Table 3 Details the columns of 'Address' table

4.2 Allergies Table

This table stores information on different types of allergies that a patient may have.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
allergy_id	int(11)	-	No	None	AUTO_INCREMENT	PRI
species	varchar(40)	-	No	None	-	
allergy_name	varchar(25)	-	No	None	-	
IUIS_allergen	varchar(25)	-	No	None	-	
type	varchar(25)	-	No	None	-	
group	varchar(30)	-	No	None	-	

length	int(11)	-	No	None	-	
GI_num	int(11)	-	No	None	-	
first_version	int(11)	-	No	None	-	

Table 4 Details the columns of the 'Allergies' table.

4.3 Drugs Table

This table stores information on different types of medications available in the market. The indication field holds information about what symptoms this drug can treat in paragraph text form. There is a primary key on drug_id column since this will be looked up frequently.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
drug_id	varchar(20)	-	No	None	-	PRI
barcode	int(11)	-	No	None	-	MUL
name	varchar(20)	-	No	None	-	
description	text	-	No	None	-	
indication	text	-	No	None	-	
pharmacology	text	-	No	None	-	

Table 5 : Details the columns of the ‘drugs’ table

4.4 Drug Brands Table

This table links drugs to brand names using the *drug_id* key. The purpose of this table is to provide a relationship between all the different brand names of drugs out there to a drug in our database.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
drug_id	varchar(20)	-	No	None	-	MUL
brand_name	varchar(30)	-	No	None	-	

Table 6 : Details the columns of the 'Drug_brands' table.

4.5 Drug Dosages Table

This table links drugs to different types of dosages using the *drug_id* key. Form field describes what form this drug exists in (e.g. cream, spray, pill, syrup, etc), the route field describes how this drug is administered to the patient. The strength

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
dosage_id	int(11)	-	No	None	AUTO_INCREMENT	PRI
drug_id	varchar(20)	-	No	None	-	MUL
form	varchar(25)	-	No	None	-	
route	varchar(25)	-	No	None	-	
strength	text	-	No	None	-	

Table 7 : Details columns of the 'drug_dosages' table

4.6 Drug Interactions Table

This table links drugs to other drugs that do not interact well with them using the *drug_id* key.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
drug_id	varchar(20)	-	No	None	-	MUL
interaction_drug_id	varchar(20)	-	No	None	-	

Table 8 : Details columns the 'drug_interactions' table.

4.7 Patient Allergies Link Table

This table links patients to allergies using the *patient_id* and *allergy_id* keys.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
patient_id	int(11)	-	No	None	-	PRI
allergy_id	int(11)	-	No	None	-	PRI

Table 9 : Details the columns of the 'patient_allergies_link' tables.

4.8 Patient Details Table

This table stores detailed information on patients. It is linked to the *patient_info* table using the *patient_id* key.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
patient_id	int(11)	-	No	None	AUTO_INCREMENT	MUL
gender	int(3)	-	No	None	-	
home_phone	int(10)	-	No	None	-	
cell_phone	int(10)	-	No	None	-	
social_security	int(9)	-	No	None	-	
marital_status	int(3)	-	No	None	-	
race	varchar(15)	-	No	None	-	
language	varchar(15)	-	No	None	-	
employment_status	int(11)	-	No	None	-	

Table 10 : Shows the detailed table structure of the 'Patient Details' table columns.

4.9 Patient Drugs Table

This table links patients to medications. This table serves as the patient's prescription. It is linked to the *drugs* table using the *drug_id* key. There is a foreign key on both columns with patient_id referencing 'Patient_info' table and drug_id field referencing 'Drugs' table.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
patient_id	int(11)	-	No	None	-	MUL
drug_id	varchar(20)	-	No	None	-	MUL

Table 11 : Shows the table structure of 'Patient Drugs' table.

4.10 Patient Info Table

This table stores general information on patients.

Column Name	Data Type	Attributes	NULL	Default Value	Extra	Key
patient_id	int(11)	-	No	None	AUTO_INCREMENT	PRI
first_name	varchar(25)	-	No	None	-	
middle_name	varchar(15)	-	No	None	-	
last_name	varchar(25)	-	No	None	-	
DOB	date	-	No	None	-	MUL
address_id	bigint(20)	-	Yes	NULL	-	

Table 12 : Show the table structure of the 'Patient Info' table.

5 Summary

5.1 Advantages of Design

The Ashvin system design has many advantages from the three-tier architecture, barcode scanning, and integration of EHR databases. The Ashvin system aims to provide consistent, accurate, and scalable access to medical records while preventing medication administration errors.

The three-tier architecture with pre-defined interfaces allows for easy unit, and integration testing during development, as well as growth, and modification to individual modules without affecting others. This allows the Ashvin system to adapt to environmental and need-based changes. Additionally, by separating the responsibilities of the project into three distinct modules, it is possible for each tier to be developed concurrently.

The barcode scanning allows for consistent access to information, since barcode protocols are well defined, and strictly followed by hospitals that will implement the Ashvin system. Using barcodes to read identify patients and medications allows hospitals to implement the Ashvin system without implementing new identification methods.

5.2 Disadvantages of Design

The Ashvin system uses a three-tier architecture, which adds complexity to the final product, as well as prevents full optimization of calls between the presentation layer and the data layer.

The Ashvin system also uses barcode scanning, which relies heavily on camera resolution, and thus is not suitable in all environments. Additionally, RFID is less battery intensive, and can be longer range, without requiring the user to look directly at the patient, or medication.

5.3 Design Rationale

The Ashvin team chose our design because it required the least effort for hospitals to implement, while maintaining scalability, and reliability.

6 Requirements Traceability Matrix

6.1 Traceability by Requirement Numbers

Requirement Number	Design Component
FR-001	
FR-002	2.6.1.1
FR-003	2.4.1.4
FR-004	2.4.1.3
FR-005	2.4.1.2
FR-006	
FR-007	
FR-008	
FR-009	
FR-010	
FR-011	2.5.2.6
FR-012	2.4.1.2, 2.4.1.5, 2.5.2.3
FR-013	2.4.1.2, 2.4.1.5
FR-014	2.4.1.2, 2.4.1.5, 2.5.2.2, 2.5.2.6

FR-015	2.4.1.2, 2.4.1.5
FR-016	2.4.1.2, 2.4.1.5
FR-017	2.4.1.2, 2.4.1.5
FR-018	2.4.1.2, 2.4.1.5
FR-019	2.4.1.2, 2.4.1.5
FR-020	2.4.1.2, 2.4.1.5
FR-021	
FR-022	2.4.1.1, 2.6.1.2

6.2 Traceability by Design Component

Design Component	Requirement Number
2.4.1.1	FR-022
2.4.1.2	FR-005, FR-012, FR-013, FR-014, FR-015, FR-016, FR-017, FR-018, FR-019, FR-020
2.4.1.3	FR-004
2.4.1.4	FR-003
2.4.1.5	FR-012, FR-013, FR-014, FR-015, FR-016, FR-017, FR-018, FR-019, FR-020
2.5.1.1	
2.5.1.2	FR-004, FR-007, FR-008, FR-009
2.5.1.3	
2.5.2.1	FR-006, FR-022
2.5.2.2	FR-014
2.5.2.3	FR-012
2.5.2.4	

2.5.2.5	
2.5.2.6	FR-011, FR-014
2.5.2.7	
2.6.1.1	FR-002
2.6.1.2	FR-022