System and Software Architecture Description (SSAD)

Cash Doctor 3.0 Mobile APP
Team 12

Name	**Primary Role**
Alisha Parvez	Life Cycle Planner
Ekasit Jarussinvichai	Requirements Engineer
Kenneth Anguka	Requirements Engineer
Kshama Krishnan	Prototyper
Le Zhuang	Feasibility Analyst
Shreya Sharma	System/Software Architect
Steven Helferich	Project Manager
Xichao Wang	Operational Concept Engineer

April 27, 2015

Version History

Date	Author	Version	Changes made	Rationale
10/13/14	Kshama Shreya	1.0	Using the SSAD template, determined the purpose and the status of the current architecture of the project and system analysis using use cases, artifacts and system context diagrams	• To depict different non-technical architectural overview of the project
10/20/14	Kshama	2.0	1. Changed terminologies so that a standard is followed across all other documents like Prototype and Requirements 2.Made changes in the functionalities so that it is consistent with Prototype and Requirement documents	• To follow a standard and avoid confusion with respect to terminologies • The architecture needs to be consistent, hence the change.
11/27/14	Kshama	3.0	Added all remaining sections	• Exit condition of Draft DCP Package
04/27/15	Ekasit	4.0	Update to align with as built application	• Deliver project

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the SSAD

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the Cash Doctor Mobile Application 3.0. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Status of the SSAD

This current status of the document provides the non-technical comprehensive architectural overview of the system, not using the different technical details as of now of Cash Doctor Mobile Application 3.0.

2.System Analysis

2.1 System Analysis Overview

The primary purpose of the Cash Doctor Mobile Application 3.0 is empowering consumers with the cost and quality of care by sharing pricing and review information of healthcare costs. Through the development of a free mobile app, we will allow consumers, globally, to share the cost of care, thus driving down the cost of medical services while improving the health of consumers. This project constitutes of three sections; section 1 will allow the consumer to capture a picture of a health care receipt via the smart phone's camera, define the charges on the
bill while attaching to a provider in a geo location (resembles the current bank mobile deposit app) Section 2 will allow the consumer to lock on to providers creating a private user network with updates and comparison features and section 3 will allow users to search data posted by other users - geolocation, provider, other users, specialty, lifestyle.

2.1.1 System Context

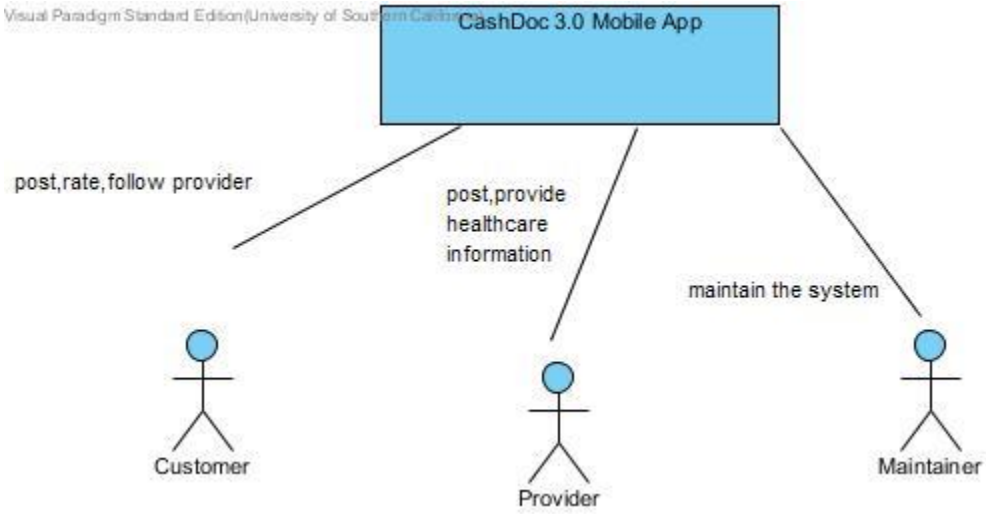


Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Consumer	A consumer is a user who can share, search compare, rate and review healthcare information. He can follow or unfollow the providers for getting updates.	1.Can register as a consumer 2.Can edit his profile page 3.Can share the prices by uploading his medical bills. 4.Can follow a provider to get updates. 5.Can rate and review providers. 6.Can search for providers based on different criteria.
Provider	Provide healthcare information like price posted, videos, photos and blogs.	1. Can register as a provider. 2.Can share prices based on the service offered.

Actor	Description	Responsibilities
Maintainer	Maintain the website by approving newly registered users.	1.Filter invalid or bad content.

2.1.2 Artifacts & Information

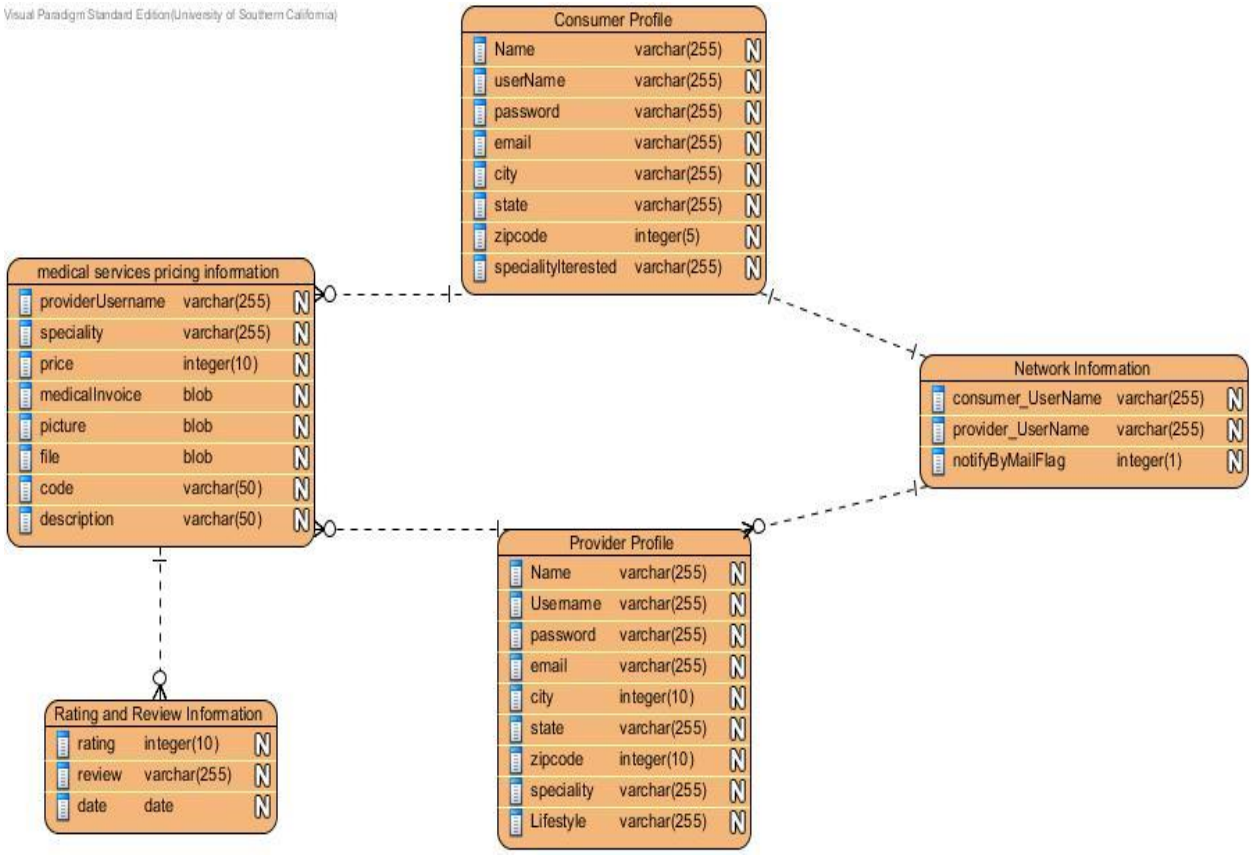


Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
ATF-1:Consumer Profile	Contains information about the username, email id, city, and state.
ATF-2:Provider Profile	Contains information about the username, email id, working company, city, state, and speciality of self.
ATF-3:Medical services prices	Contains information about the price of the medical bills, the medical invoices uploaded, files, pictures etc.
ATF-4:Ratings and review	Contains the rating and the reviews written by the consumers about the healthcare providers
ATF-5:Network Information	Contains information about which consumer has subscribed for which providers' information.

2.1.3 Behavior

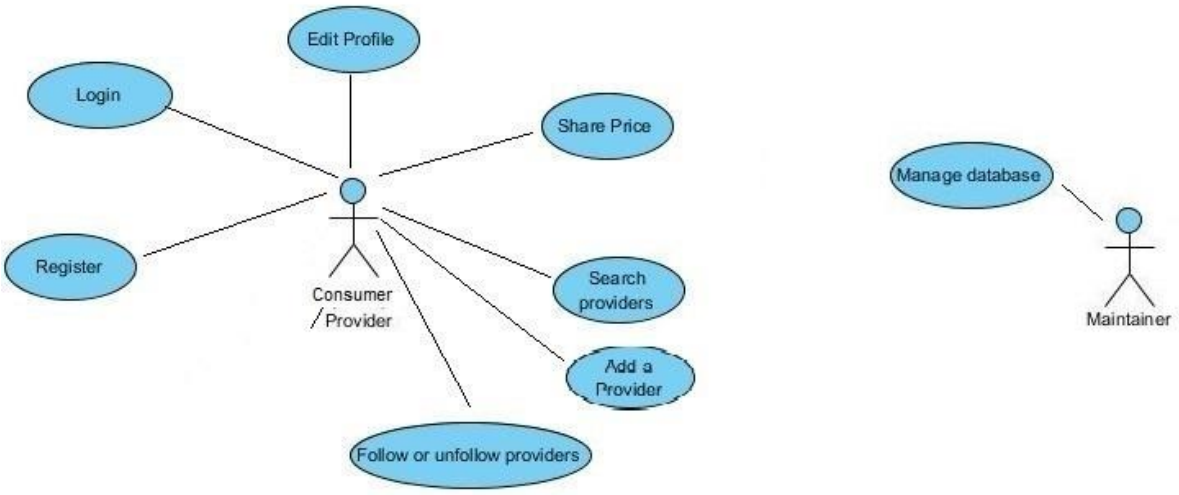


Figure 3: Process Diagram

Table 3: USE-CASE GRID

ID	NAME	PRIMARY ACTORS
UC1	Register/Login to Consumer profile	Consumer/ Provider
UC2	View Dashboard Page	Consumer/ Provider
UC3	Edit Consumer Profile	Consumer/ Provider
UC4	Share Prices	Consumer/ Provider
UC5	Search Provider	Consumer/ Provider
UC6	Follow/Unfollow a Provider	Consumer/ Provider
UC7	Rate a provider	Consumer/ Provider
UC8	Add New Provider	Consumer/ Provider
UC9	Maintain the system	Maintainer

2.1.3.1 Capability

2.1.3.1.1 Create/Login consumer profile, Create/Login Provider profile

Identifier	UC1:Login
Purpose	Determines if a user logging in to the system can be authenticated, and, if so, what the person's privileges are as a user of the system, i.e., what the person is authorized to do when using the system
Requirements	WC_3086: As a consumer I can register as a user.
Development Risks	None
Pre-conditions	CMS database should be initialized
Post-conditions	If user is authorized s/he is given access to “My accounts” page; otherwise, s/he is denied access to the system.

Typical Course of Action:Login-Successful

Seq#	Actor’s Action	System’s Response
1	[User] Enters a user name and password	
2	[User] Clicks Login button	
3		Sends username and password to the CMS server
4		[valid] Redirects the actor to “My Accounts” Page

Alternate Course of Action:Login-New Account

Seq#	Actor’s Action	System’s Response
1	User click “Register New Account” link	
2		Directs to the “New Account” Page
3	User inputs all the fields and clicks register	
4		[valid]Directs to the “My Account” Page
		[invalid]Displays error messages

Alternate Course of Action:Login-Failure

Seq#	Actor's Action	System's Response
1-3	Refer to Typical Course of Action	
4		Displays An error message: "username or password is wrong" in a dialog box
5	Click OK button	
6		Redirects the user to the login page

Alternate Course of Action: Login: Forgotten Password

Seq#	Actor's Action	System's Response
1	User click "Forgotten username or password" link	
2		Sends An email message, providing the details of the user's credentials to his email id.
3	User opens the email and inputs the username and password again	
4		Refer to Typical Course of Action

2.1.3.1.2 View Dashboard Page

Identifier	UC2
Purpose	To provide a dashboard for the consumer wherein the consumer gets to know all the features available to him like price share, compares prices, searching various healthcare services, follow providers
Requirements	WC_3087: As a consumer I can access my existing account by user ID and password, I can view my existing dashboard.
Development Risks	None
Pre-conditions	The user should be logged in
Post-conditions	The user can explore all the features available to him in his account

Typical Course of Action

Seq#	Actor's Action	System's Response
1	Consumer clicks on any feature that he wants to explore	
2		System directs him to the required page
3	Consumer modifies information in his dashboard	
4		[valid]Modified information is updated in the server [invalid]Error message is displayed.

2.1.3.1.3 Edit Consumer Profile

Identifier	UC3
Purpose	Allows the consumer to change his information
Requirements	None
Development Risks	None
Pre-conditions	Consumer clicks "Edit profile" from "My accounts" Page
Post-conditions	After consumer click "Save change", their information will be changed according to data in this page.

Typical Course of Action

Seq#	Actor's Action	System's Response
1	Consumer modifies information in his dashboard	
2		[valid]Modified information is updated in the server [invalid]Error message is displayed.

2.1.3.1.4 Share Prices

Identifier	UC4
Purpose	Uploading medical bills so that the cost of that particular healthcare service is known. The uploading can be done manually or through an OCR capture
Requirements	WC_3083: an individual consumer can manually enter price information for sharing. WC_3082: An individual consumer can capture an image and code an invoice for sharing.
Development Risks	OCR is not accurate and will not recognize handwritten medical bills
Pre-conditions	UC1,UC2
Post-conditions	If the price and other fields are valid, the information is updated in the database.

Typical Course of Action: Share Price manually

Seq#	Actor's Action	System's Response
1	Clicks share price manually	
2		The provider's name, the service offered and the price field is displayed on screen
3	The user input all the fields	
4		[valid]stores information into the database [invalid]Displays error message

Alternate Course of Action:Share Price through OCR capture

Seq#	Actor's Action	System's Response
1	Clicks share price through photo capture	
2		The photo option is enabled
3	Consumer clicks the photo and uploads to server	
4		If OCR reads the medical service and corresponding medical price correctly, updates in database Else, makes the user to try again.

2.1.3.1.5 Search Provider

Identifier	UC5
Purpose	Consumers can search doctors based on multiple conditions like locality, price, service etc
Requirements	WC_3084: An individual consumer can search for healthcare pricing, provider by location, price, code, and specialty.
Development Risks	None
Pre-conditions	UC1,UC2
Post-conditions	The consumer will be able to view a list of all the doctors and their price based on multiple conditions specified.

Typical Course of Action

Seq#	Actor's Action	System's Response
1	Inputs the fields-locality, service and price and clicks search	
2		[valid]retrieves all the records matching the condition [invalid]Displays an error message

Alternate Course of Action

Seq#	Actor's Action	System's Response
1-1	Refer to typical course of action	
2		If no records found, then displays "No records found"

2.1.3.1.6 Follow/Unfollow a Provider

Identifier	UC6
Purpose	My network page will display list of providers whom users are following. List will state briefly information about updated data. Users can click each provider to go to provider page of that provider for more information.
Requirements	WC_3088: As a consumer I can create a private network and join

	existing networks.
Development Risks	None
Pre-conditions	UC1,UC2
Post-conditions	The user creates a new network

Typical Course of Action

Seq#	Actor's Action	System's Response
1	Consumer searches the provider whom he wants to follow	
2		Displays all information based on his search criteria
3	The consumer clicks "follows/unfollows" for that provider	
4		The system pops up asking if the consumer would want to subscribe for emails from the provider.
5	The consumer ticks the appropriate data.	
5		The information about the network that the user wants too join is updated in the database

2.1.3.1.7 Consumer can rate a provider

Identifier	UC7
Purpose	The user has the ability to rate a provider
Requirements	None
Development Risks	WC_3091: As a consumer I can rate a provider.
Pre-conditions	UC5
Post-conditions	None

Typical Course of Action

Seq#	Actor's Action	System's Response
1	Searches for the provider to whom the review/rating needs to be written	
2		Retrieves the provider details
3	Writes the rating	
4		The information is updated in the database.

2.1.3.1.8 Add New Provider

Identifier	UC8
Purpose	Search page can add new provider to the system if users cannot find a particular provider in search result
Requirements	WC_3084: An individual consumer can search for healthcare pricing, provider by location, price, code, and specialty.
Development Risks	None
Pre-conditions	UC5
Post-conditions	New provider is added so user can share the price to that provider

Typical Course of Action

Seq#	Actor's Action	System's Response
1	User click "add provider" button in search result	
2	User fill provider's information and submit	
3		System create new provider and redirect user to that provider page

2.1.3.1.9 Maintain the system

Identifier	UC9
Purpose	The maintainer is responsible for approving the new creation requests
Requirements	None
Development Risks	None
Pre-conditions	UC1
Post-conditions	The new users are approved by the maintainer

Typical Course of Action

Seq#	Actor's Action	System's Response
1	Checks for new requests	
2		List of new requests are retrieved
3	Maintainer approves the requests that are valid and unapproves the invalid ones	
4		The information is updated into the database

2.1.4Modes of Operation

The CashDoctor 3.0 application operates in one mode. So there is no additional information concerning modes of operations need to be provided.

2.2 System Analysis Rationale

Some of the features that can be misunderstood and explanations for the features are given below:-

- There are 2 kinds of users in our application, Consumers and Providers. Consumers are users who shares and search healthcare information. **Providers can be doctors and even healthcare centers who can provide the service to the user community**

- The consumer can follow his preferred list of providers and also subscribe to the provider feeds by email. Note that network here essentially means a consumer following or unfollowing a provider. There is no feature in the application that allows many consumers and providers to be part of one network.

- When the invoice is being upload by the consumer, it may be the case that the OCR software would not be successful in detecting the invoice contents, in that case, the consumer will have an option to upload the invoice directly onto the database.

3. Technology-Independent Model

This document has no Technology-independent model because our client already have a website in place and they want us to build the mobile phone application that is equivalent. Hence the technologies are finalized based on the client's requirements and therefore no need for the Technology-independent model.

4.Technology-Specific System Design

4.1 Design Overview

4.1.1 System Structure

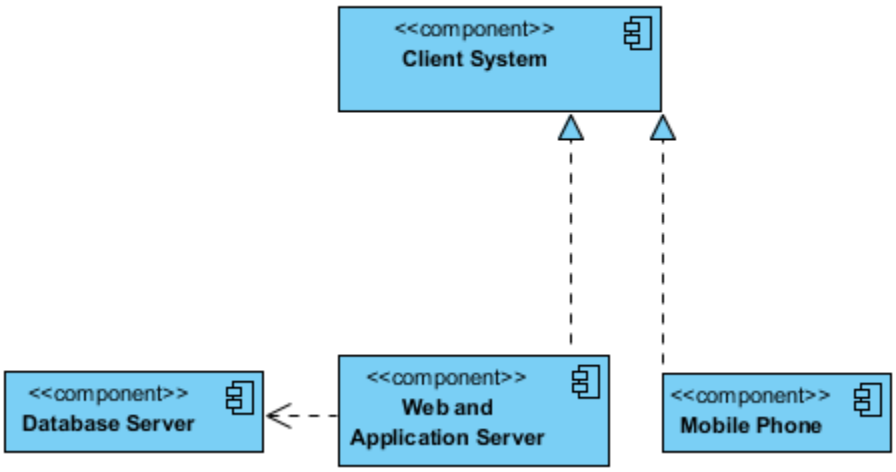


Figure 4 : Hardware Component Class Diagram

Table 5: Hardware Component Description

Hardware Component	Description
Mobile Phone	Can be any Android or iPhone mobile phone
Web and Application server	Web and Application server handles the JSON calls and the business logic respectively.
Database server	A server where the cashdoctor database is running

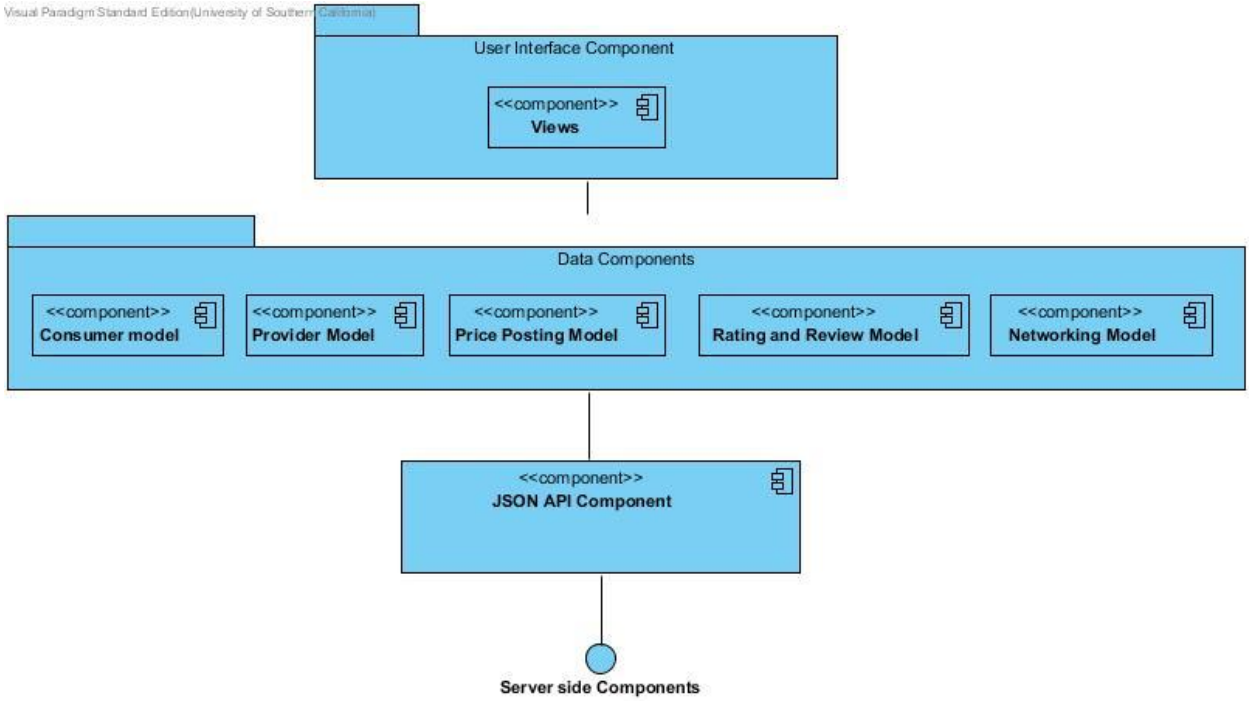


Figure 5: Software Component Class Diagram

Table 6: Software Component Description

Software Component	Description
User Interface Components	Contains the views that covers all the features of the application
Data Components	Contains the models for each of the features
JSON API Component	Server side for handling JSON request-response objects
Server side Components	**Since the client already has backend code written, using Servlets in J2EE environment and Apache Tomcat currently, there is no reason to specify the details in this document.**

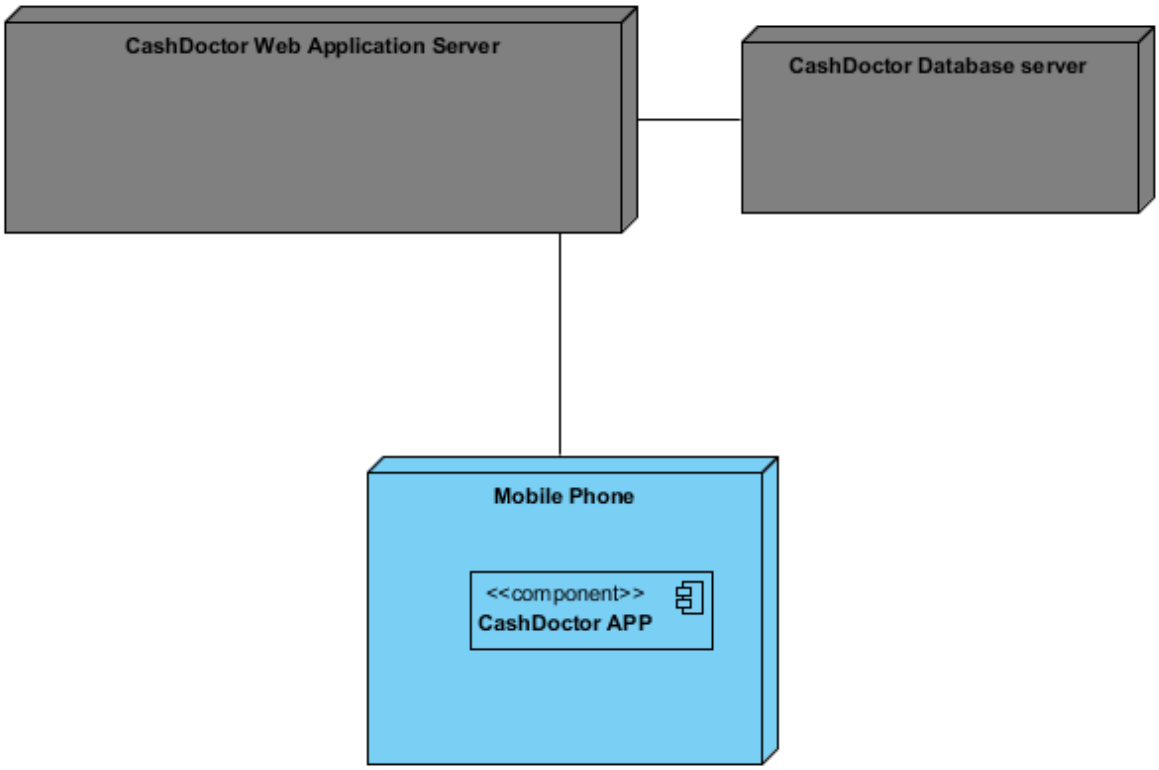


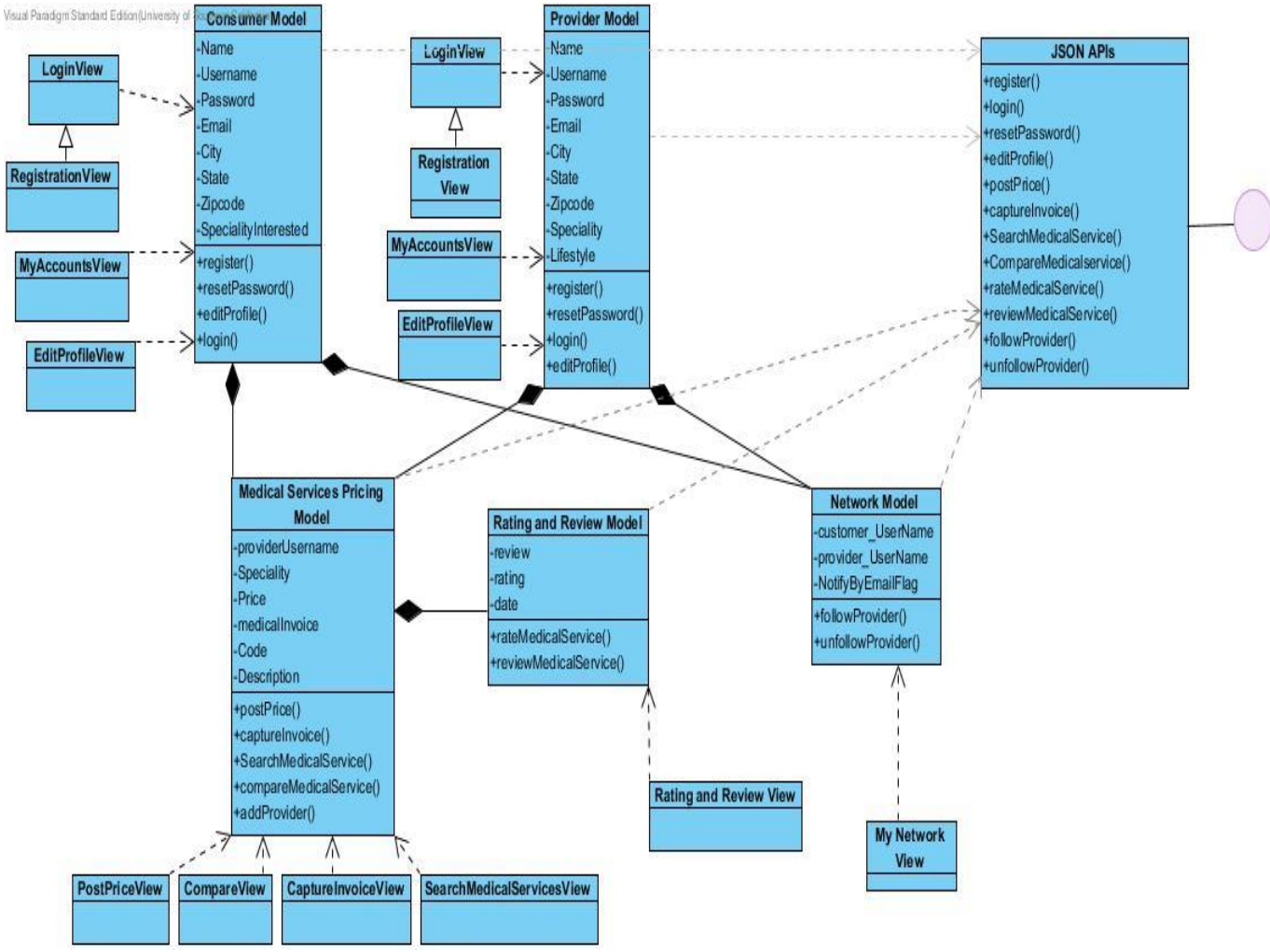
Figure 6: Deployment Diagram

Table 7:Software Deployment Description

Support Software Component	Description
Mobile Phone	The CashDoctor APP is available for Android and OS X phones.
CashDoctor Web Application Server	The client currently has a J2EE environment to take up request response object. Tomcat server is currently being used.
Cash Doctor Database server	The client currently uses SQL server to maintain the website application.

4.1.2 Design Classes

Figure 7: Design Class Diagram



Note: Since the client has already implemented the server side coding, the design diagram includes only the JSON APIs that are called. It does not include the details of the database connectivity.

4.1.3 Process Realization

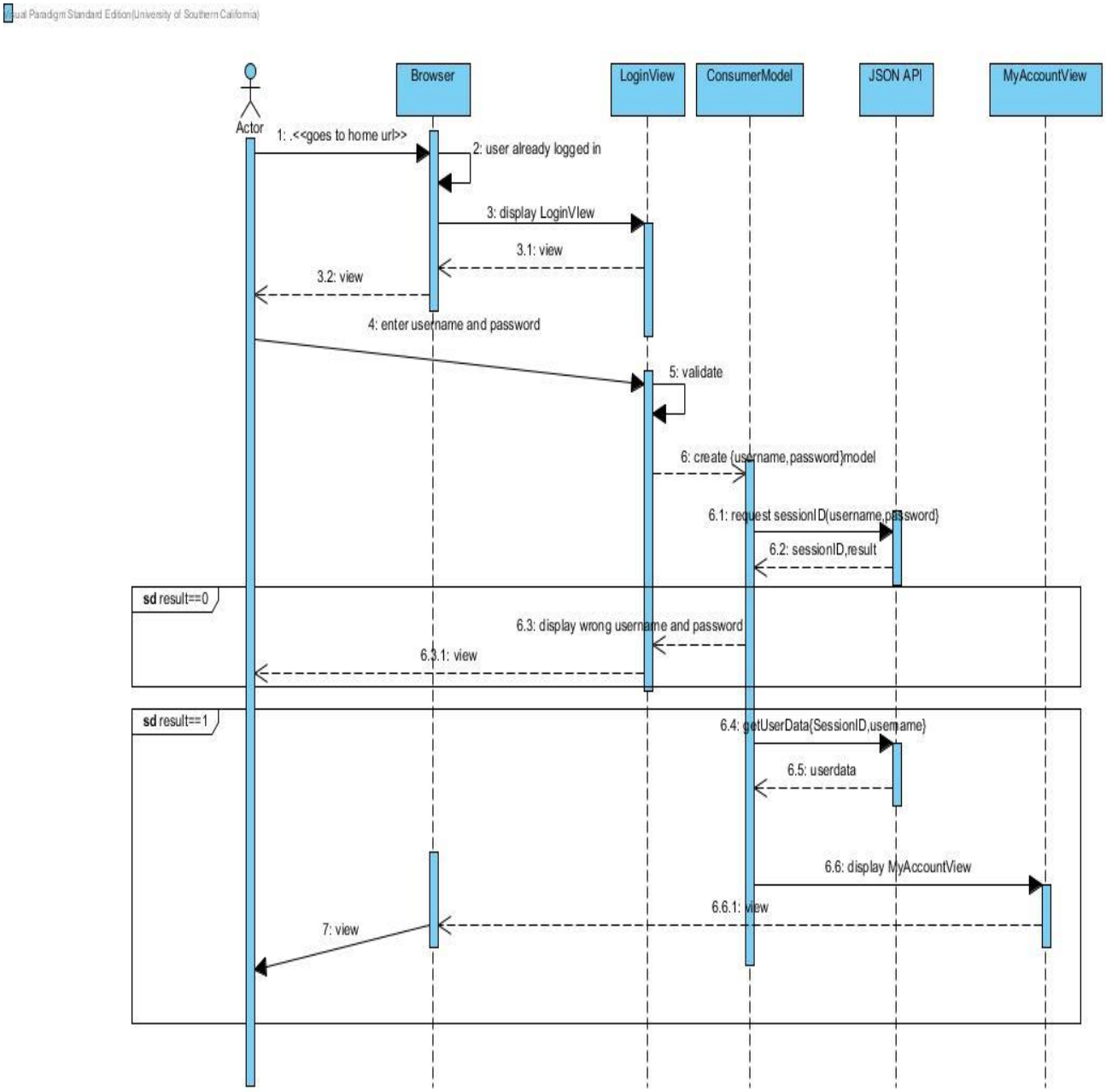


Figure 8: Process Realization Diagram for login sequence

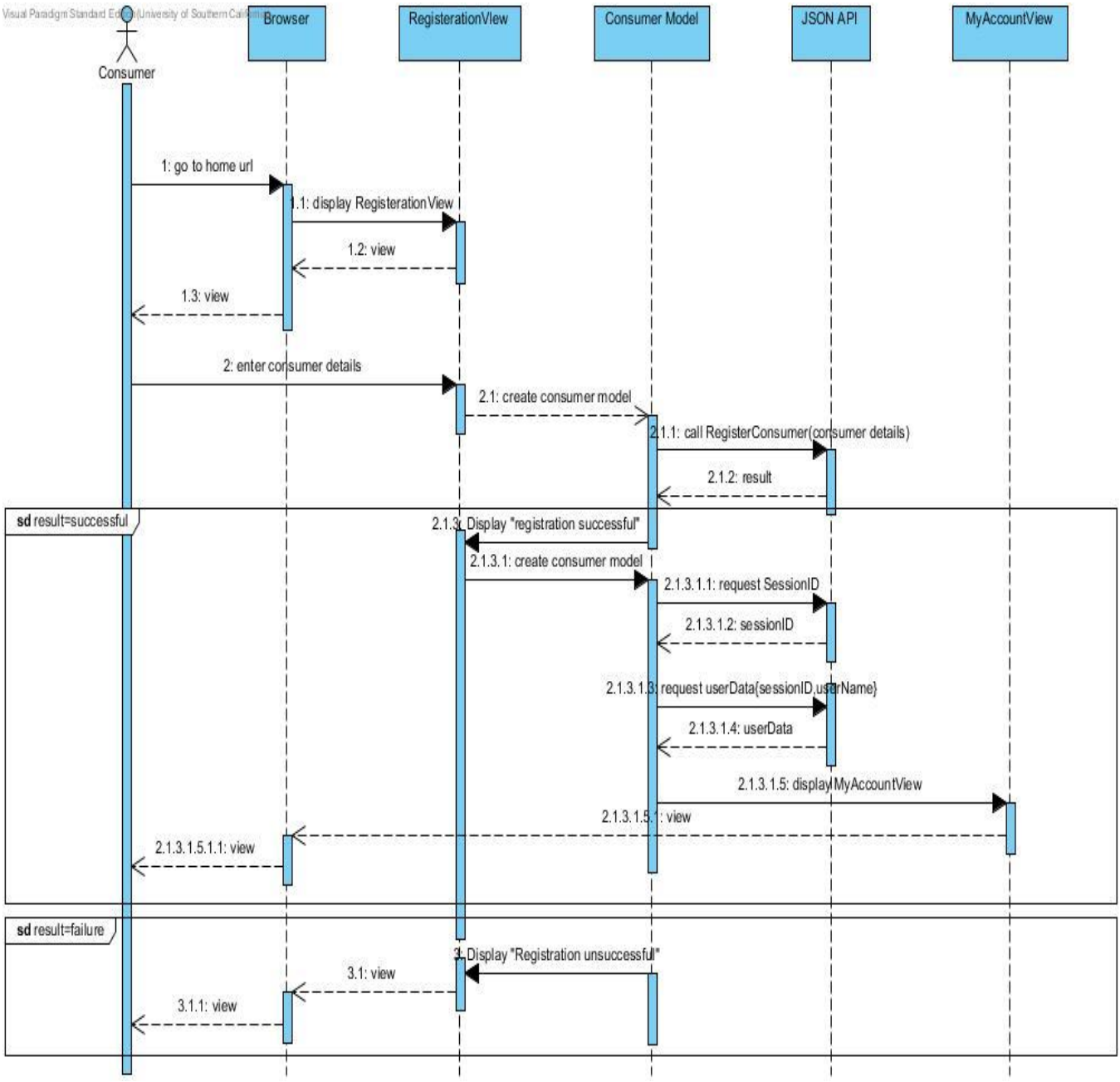


Figure 9: Process Realization Diagram for Consumer Registration

4.2 Design Rationale

The CashDoctor App already has a website equivalent, so more focus is given of the user interface of the mobile application.

- In the Hardware Component diagram, we have an application and web server in place (Tomcat) and also a database server(SQL Server) currently. In this project, we are supposed to build the mobile application so that information can be integrated with the client server. Since the servers used by the clients can be subjected to change, we do not represent in the Hardware Component Diagram.

- In the software component diagram, we will be using Backbone.js to handle the client-side code.
- We have Views that act as the user interface, the Models are divided to handle different functionalities in the system like medical price information, network information etc. **The server side components currently consists of java servlets to handle request-response objects. The reason why it is not mentioned in the diagram is because the technologies currently used by our clients are subject to change.**

- The entire coding of our project is done using Apache Cordova, which helps us to port the HTML/CSS/JS written code to different platforms. The mobile app that we will be developing will interact with the application server of the client only using the JSON API calls. The application and database servers that will be deployed by our client is not mentioned in the diagram.

- The design class diagram consists of all the features abstracted as models and having corresponding views. All the above models have functions that make JSON calls. Note that the flow ends at the JSON API because the client has given us ready-made APIs and we are not aware of the data manipulation that takes place after the JSON call.

5. Architectural Styles, Patterns and Frameworks

Name	Description	Benefits, Costs, and Limitations
Backbone.js	Client-side JavaScript Model-View framework	Benefits: 1.Gives structure to client-side code 2.Client-side code can switch views immediately without a round trip to server 3.JSON support 4.Free Limitations: 1.It could be argued that Angular.js is becoming a more popular option 2.Learning curve for team members without experience
Apache Cordova	Mobile development framework	Benefits: 1. Rapid testing and deployment; 2. Access basic native functionality; 3. Relatively flat learning curve. Limitations: 1.Poor performance especially in the automation, transition and tabs change etc.
Bootstrap	Provide front-end layout implementation development framework	Benefits: 1. Provides a clean and elegant GUI 2. Free Limitations: 1. Likely to have namespace bugs