**Hospital**

**Management Project**

Spring Semester 2017

CSC 4350, 5:30-7:15

24 April 2017

**Big Hero 5**

Gautam Ravichandran

Victoria Green

Ashwin Nair

Karankumar Parikh

Young Jun Son

Table of Contents:

Introduction………………………………………………………………………………..………3

Required Tractability Matrix…………………………………………………………………..4 - 7

System Analysis & Design…………………………………………………………………...8 - 12

Use Case…………………………………………………………………………………………. 8

Sequence Diagram…………………………………………………………………………….9 -12

Object Design & Rationale………………………………………………………………… 13 -14

Test Cases………………………………………………………………………………...…15 - 18

Rational…………………………………………………………………………………………. 19

Constructive Cost Model………………………………………………………………………...20

Project Legacy………………………………………………………………………………….. 21

Work Structure Document……………………………………………………………..…….......21

Gantt Chart.…………………………………………………………………………………….. 22

Dictionary………………………………………………………………………………….…… 23

User Guide…………………………………………………………………………………………………………………………………………... 24

Resumes…………………………………………………………………………………………………………..……..… Attached In GitHub

Introduction:

The Hospital Management System, created by Big Hero 5, is a Java program which allows the user to assist with essential tasks which occur at a hospital. Functions include the login screen, entering patient information, controlling inventory, scheduling appointments, and invoicing. The team utilized SQLite to manage its database and integrated it into the Java program. Using Github, Big Hero 5 was able to upload deliverables, code, and general project materials for easy access. The primary source of communication was through Google Hangouts. Video calls, text messages, and image-sharing led to clearly-illustrated ideas as to which direction the project was headed. After approximately four months of the development process, Big Hero 5 is able to present a working model of the Hospital Management System.

**Hospital Management System (HMS)**

**HMS- 1.0 INTRODUCTION**

A Hospital Management System (HMS) is designed to automate and organize various day-to-day activities taking place in a hospital. It also stores patient and staff database for quick future access.

**HMS- 2.0 APPOINTMENTS**

**2.0.1** The System shall help in setting up patient appointments by considering each individual case and the schedule of assigned doctor.

**2.0.2** The System shall also give doctors an option to set up or cancel an appointment.

**HMS- 2.1 APPOINTMENT DATA REQUIRED**

**2.1.1** For each patient making an appointment, the system shall request the following information:

* Patient First and Last name
* Patient ID
* Patient Age
* Doctor
* Date
* Time
* Contact Number
* Reason for Visit

**HMS- 3.0 PATIENT RECORDS**

**3.0.1** The System shall maintain patient records efficiently. This information will be manually recorded and inputted into the system after the patient’s visit to the doctor.

**3.0.2** Such records shall be accessed by assigned doctors in the future.

**HMS- 4.0 STAFF RECORDS**

**4.0.1** The System shall maintain a record of doctor appointments, prescriptions given by the doctor and details of the doctor.

**4.0.2** The System shall also maintain relevant information about the nurses and other hospital staff.

**HMS- 4.1 STAFF STATIC DATA REQUIRED**

**4.1.1** For each hospital employee, the system shall request the following information:

* First and Last Name
* Staff Type
* Date of Birth
* Age
* Contact Number
* Gender

**4.1.2** This static data shall be obtained and documented when the staff member is hired.

**HMS- 4.2 STAFF ACTIVE DATA REQUIRED**

**4.2.1** For each hospital employee, the system shall request the following information:

* List of appointments

**4.2.2** This active data shall be consistently updated when changes are made.

**HMS- 5.0 PHARMACY**

**5.0.1** The System shall efficiently maintain medicine inventory and notify when a new order needs to be placed. It keeps track of Medicine ID, medicine name, quantity, and supplier.

**HMS- 5.1 REORDER SUPPLY**

For each item in 6.0, the system immediately reacts to the following situation:

* **5.1.1** When the count of an item reaches <=300, the system shall send a notification to the user with the name of the item and the number on hand.

**HMS- 6.0 INVOICE MANAGEMENT**

**6.0.1** The System shall manage patient invoices and transaction details. It keeps track of patient ID, patient name, contact number, doctor assigned, and total cost.

**HMS- 6.1 INVOICE CREATION**

For each invoice created, the system immediately reacts to the following situation:

* **6.1.1** When the doctor finishes recording procedures and medicine administered to the patient into the system, an invoice shall be generated.
* **6.1.2** The generated invoice shall add up the cost of everything administered at the appointment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entry #** | **Paragraph #** | **System Specification Text** | **Type** | **Build** |
| 2.0.1 | 2.0 | The System shall help in setting up patient appointments by considering each individual case and the schedule of assigned doctor. | SW | B1 |
| 2.0.2 | 2.0 | The System shall give doctors an option to set up or cancel an appointment. | SW | B1 |
| 2.1.1 | 2.1 | For each patient making an appointment, the system shall request the following information: Patient First and Last name, age, patient ID, doctor, date, time, contact number, reason for visit. | SW | B1 |
| 3.0.1 | 3.0 | The System shall maintain patient records efficiently. | SW | B2 |
| 3.0.2 | 3.0 | Such records shall be accessed by assigned doctors in the future. | SW | B2 |
| 4.0.1 | 4.0 | The System shall maintain a record of doctor appointments, prescriptions given by the doctor and details of the doctor. | SW | B2 |
| 4.0.2 | 4.0 | The System shall maintain relevant information about the nurses and other hospital staff. | SW | B2 |
| 4.1.1 | 4.1 | For each hospital employee, the system shall request the following information: first and last name, gender, date of birth, age, contact number, staff type. | SW | B2 |
| 4.1.2 | 4.1 | This static data shall be obtained and documented when the staff member is hired. | SW | B2 |
| 4.2.1 | 4.2 | For each hospital employee, the system shall request the following information: list of appointments. | SW | B2 |
| 4.2.2 | 4.2 | This active data shall be consistently updated when changes are made. | SW | B2 |
| 5.0.1 | 5.0 | The System shall efficiently maintain medicine inventory and notify when a new order needs to be placed. It keeps track of Medicine ID, medicine name, quantity, and supplier. | SW | B3 |
| 5.1.1 | 5.1 | When the count of an item reaches <=300, the system shall send a notification to the user with the name of the item and the number on hand. | SW | B3 |
| 6.0.1 | 6.0 | The System shall manage patient invoices and transaction details. It keeps track of patient ID, patient name, contact number, doctor assigned, and total cost. | SW | B4 |
| 6.1.1 | 6.1 | When the doctor finishes recording procedures and medicine administered to the patient into the system, an invoice shall be generated. | SW | B4 |
| 6.1.2 | 6.1 | The generated invoice shall add up the cost of everything administered at the appointment. | SW | B4 |

System Analysis & Design:

Hospital Management System Use Case Diagram:



Login Sequence Diagram:

C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\LoginSeqDia.png

New User Registration Sequence Diagram:

C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\NewUseRegSeqDia.png

Schedule Appointment Sequence Diagram:

C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\SchedAptSeqDia.png

Doctor Appointment Sequence Diagram:

C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\DocAptSeqDia.png

Pharmacy Sequence Diagram:

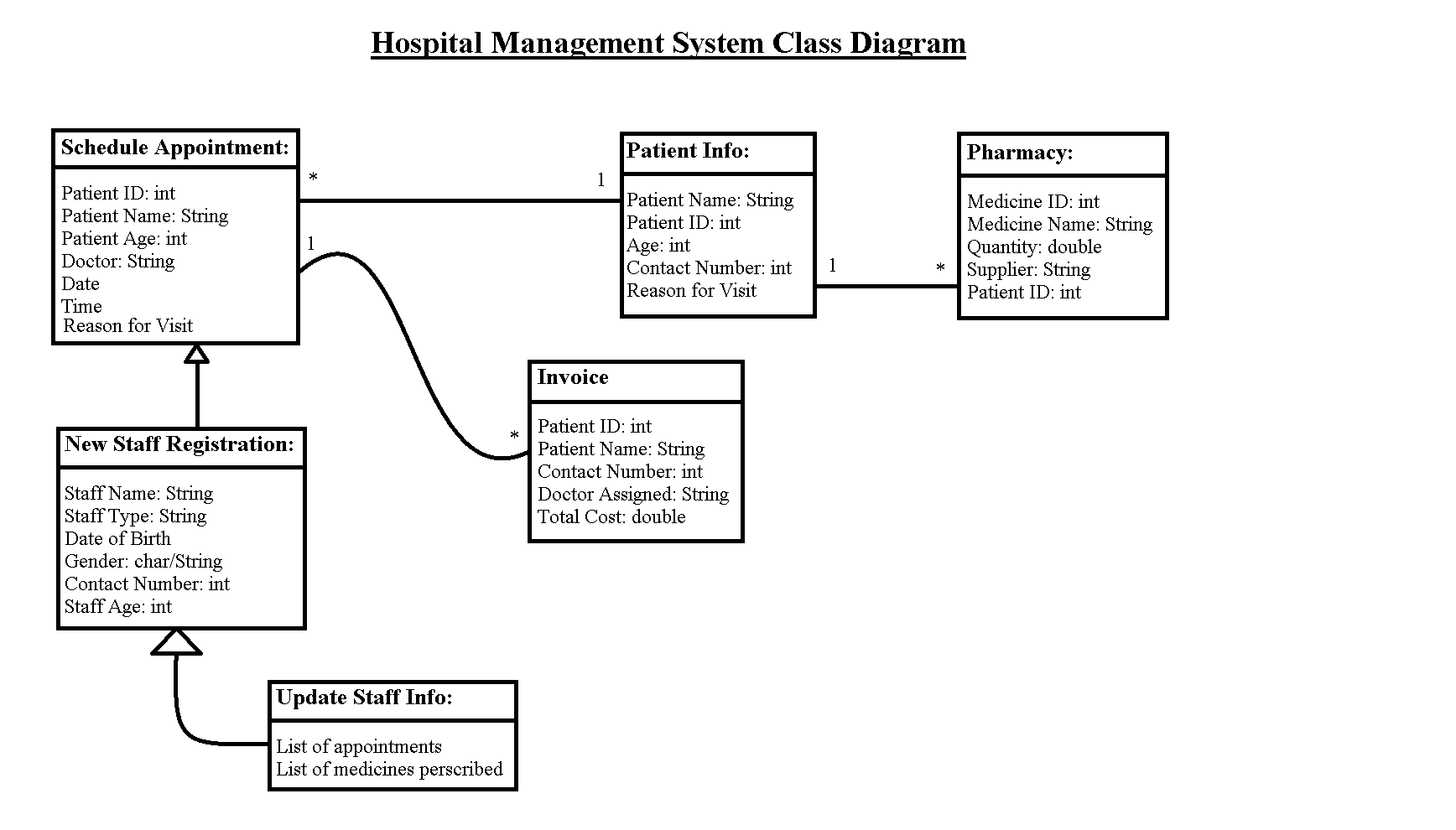
C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\Pharmacy1.png

C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\Pharmacy2.png

C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\Pharmacy3.png

Invoice Sequence Diagram:

C:\Users\Gautam\Documents\GitHub\HospitalManagement\Presentation\PresImage\InvSeqDia.png



**Application Objects Rationale:**

**Schedule Appointment**

* *Patient ID:* Required for identification of each patient in the database
* *Patient Name:* Required information for scheduling and documentation purposes
* *Doctor:* Required information so patient gets the doctor they want
* *Date:* Required information for patient and doctor for scheduling purposes
* *Time:* Required information for patient and doctor for scheduling purposes
* *Reason for Visit:* Useful information for the doctor to know before appointment

**New Staff Info**

* *Staff ID:* Required for identification of staff member in the database
* *Staff Name:* Required information for documentation purposes
* *Staff Type:* Whether the staff is a doctor or other member, useful for update staff info
* *Date of Birth:* Useful information for patients to have when choosing a doctor
* *Age:* Useful information for patients to have when choosing a doctor
* *Gender:* Useful information for patients to have when choosing a doctor
* *Contact Number:* For other staff members to reach this staff member

**Update Staff Info**

* *List of Appointments:* Required so there is no overlap in scheduling appointments
* *List of Medicines Prescribed:* Useful information to have, some patients like to know

**Patient Info**

* *Patient Name:* Required information for scheduling and documentation purposes
* *Patient ID:* Required for identification of each patient in the database
* *Age:* Useful information for doctors to know when treating patients
* *Contact Number:* Required in case staff members need to reach the patient
* *Reason for Visit:* Useful information for the doctor to know before appointment

**Pharmacy**

* *Medicine ID:* Required information for identification of medicine in the database
* *Medicine Name:* Required information for documentation purposes
* *Quantity:* Required information in order to maintain supply
* *Supplier:* Required information for documentation purposes
* *Patient ID:* Required for identification of each patient taking medicine

**Invoice**

* *Patient ID:* Required for identification of each patient with an invoice
* *Patient Name:* Required information for invoice documentation purposes
* *Contact Number:* Required in case staff members need to reach the patient
* *Doctor:* Required for invoice documentation purposes
* *Total Cost:* Required value on the invoice for the patient

**Software Architecture Rationale**

The intent of the hospital management system is to obtain and update information for staff, as well as maintain several services for multiple patients at any given time. As a result, it has been determined that a client-server model is the best software architecture for this project.

**Test Cases**

**TEST 1**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | User Login |
| Tester | Ashwin Nair |
| Input | User enters the login credentials – username and login. |
| Action | If the login credentials are accurate, and if the username and password match, then the user will be deemed logged in and will have access to the user interface of the Hospital Management System. |
| Test Type | Acceptance Test, System Test |

**TEST 2**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Scheduling an Appointment |
| Tester | Ashwin Nair |
| Input | The user must provide:   * Patient Name (First and Last) * Patient ID * Doctor * Date * Time * Contact (Phone) * Visit Reason |
| Action | The system will process the information and schedule the appointment time according to staff availability. |
| Test Type | Acceptance Test, System Test |

**TEST 3**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Medicine Information |
| Tester | Ashwin Nair |
| Input | User selects medicine ID from the dropdown selection. |
| Action | The system returns the medicine’s Name, Supplier, and Quantity |
| Test Type | Acceptance Test, System Test |

**TEST 4**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Medicine Low Quantity Alert |
| Tester | Ashwin Nair |
| Input | User selects medicine, |
| Action | Medicine ID selection is rendered, and if the medicine quantity drops below the minimum of 300, the alert box notifies the user that the item is low in quantity |
| Test Type | Acceptance Test, System Test |

**TEST 5**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Medicine Reorder |
| Tester | Ashwin Nair |
| Input | User enters the quantity of the medicine needed to be reordered. |
| Action | Medicine Totals are updated in the database upon entering the quantity. |
| Test Type | Acceptance Test, System Test |

**TEST 6**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Creating New Staff Information |
| Tester | Ashwin Nair |
| Input | User fills out information for the following criteria:   * First and Last Name * Staff Type * Age * Contact (Phone) * Gender |
| Action | This staff member’s information is added in the database and rendered on log as a new staff member. |
| Test Type | Acceptance Test, System Test |

**TEST 7**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Updating Staff Information |
| Tester | Ashwin Nair |
| Input | User updates necessary information from the following criteria:   * First and Last Name * Staff Type * Age * Contact (Phone) * Gender |
| Action | This staff member’s information is updated in the database and rendered on log. |
| Test Type | Acceptance Test, System Test |

**TEST 8**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Creating New Patient Information |
| Tester | Ashwin Nair |
| Input | User fills out information for the following criteria regarding the patient:   * First and Last Name * Age * Contact (Phone) * Address * Gender |
| Action | This patient’s information is added in the database and rendered on log as a new patient in the server. |
| Test Type | Acceptance Test, System Test |

**TEST 9**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Attribute Name | Updating Patient Information |
| Tester | Ashwin Nair |
| Input | User fills out information for the following criteria regarding the patient:   * First and Last Name * Age * Contact (Phone) * Address * Gender |
| Action | This patient’s information is updated in the database and rendered on log. |
| Test Type | Acceptance Test, System Test |

Rationale:

Baymax Hospital is a rising medical center located in the center of Atlanta, Georgia. In order to help deliver the most advanced and compassionate care, our management system must be able to efficiently obtain, update, and maintain large amounts of information for the hospital’s staff and patients. It will allow quick scheduling of appointments, maintain a healthy supply of pharmaceuticals, and create pharmacy invoices for our patients.

The Hospital Management System will be linked to a database and must comply with all of Baymax Hospital’s requirements. The system will be loaded onto the desktop of both the hospital front desk for check-in purposes, as well as the front desk of the pharmacy. This software is strictly for use by hospital staff in order to better serve our patients.

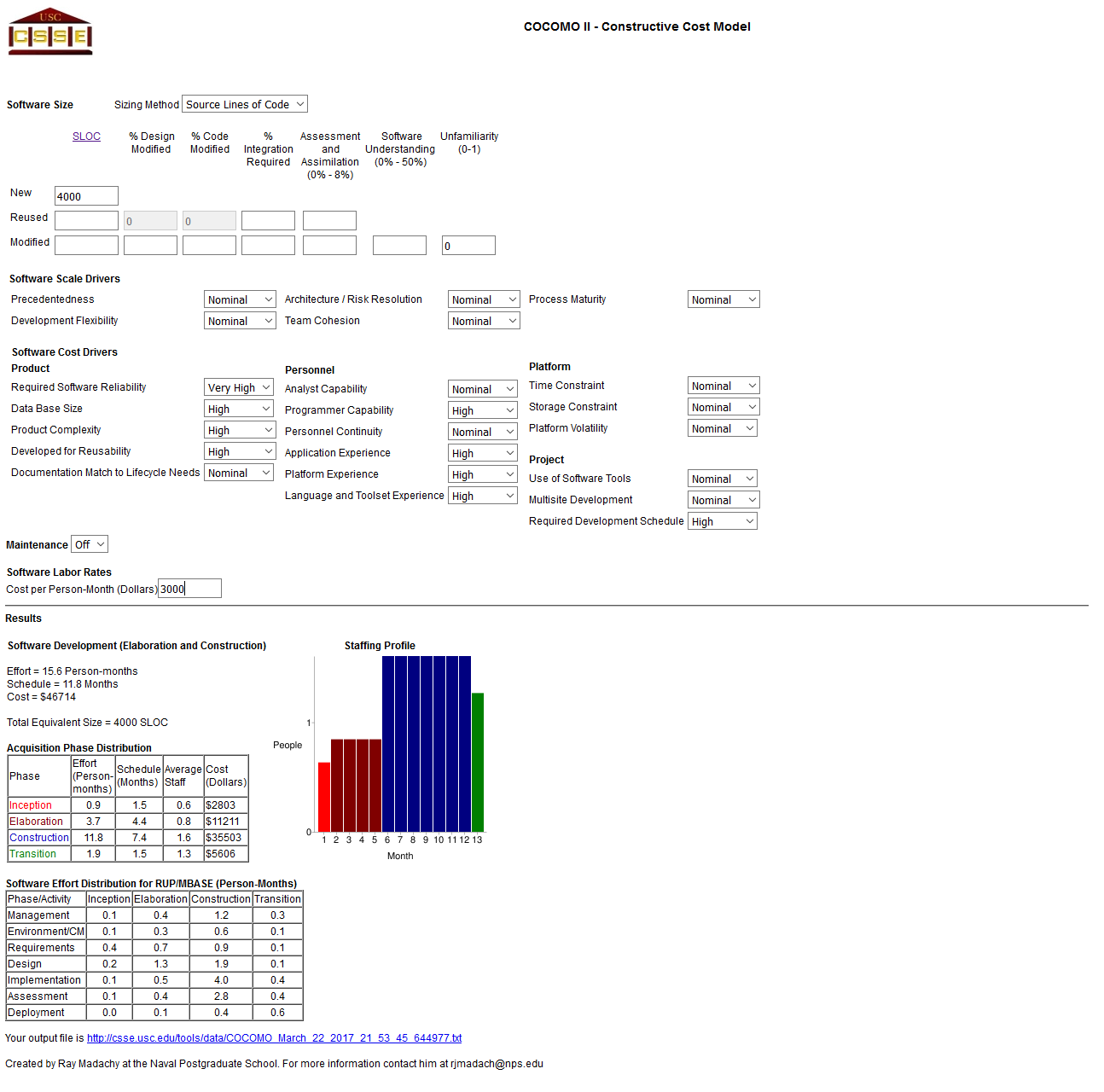
When the software is loaded and the receptionist logs in, they will first be greeted with the option to schedule an appointment for patients that are already in the database. The patient and the doctor that is being requested are selected, as well as a date, time, and reason for visit. When the appointment is scheduled, it is entered into the database to be accessed later. Patient history shows each patient’s name, age, contact number, and the information for the upcoming appointment is shown. There is also functionality to view all scheduled appointments for each doctor in the hospital.

The system also maintains the hospital’s pharmacy. If a specific medication’s quantity drops below 300, the front desk of the pharmacy is prompted to resupply the medicine. The quantity is immediately checked upon entering the pharmacy module, as well as every time a medication is purchased. When the receptionist sells medication to a patient, an invoice is generated that obtains all purchased medications and calculates the total price. All information is stored and updated in real-time.

The information provided above is in line with all of Baymax Hospital’s requirements specified by Big Hero 5, before the system was produced.

1. The login page allows the receptionist to log into the system using a correct username and password.
2. The Schedule Appointment module allows the receptionist to book an appointment during an available time slot.
3. The Doctor Appointments module displays all appointments made for a specified doctor.
4. Patient Information maintains each patient’s name, age, contact number, and upcoming appointments.
5. The pharmacy maintains the medicine inventory, notifies the receptionist when a quantity drops under 300, and distributes prescribed medications to patients.
6. An invoice is generated when a medication is sold to the patient.

Constructive Cost Model:



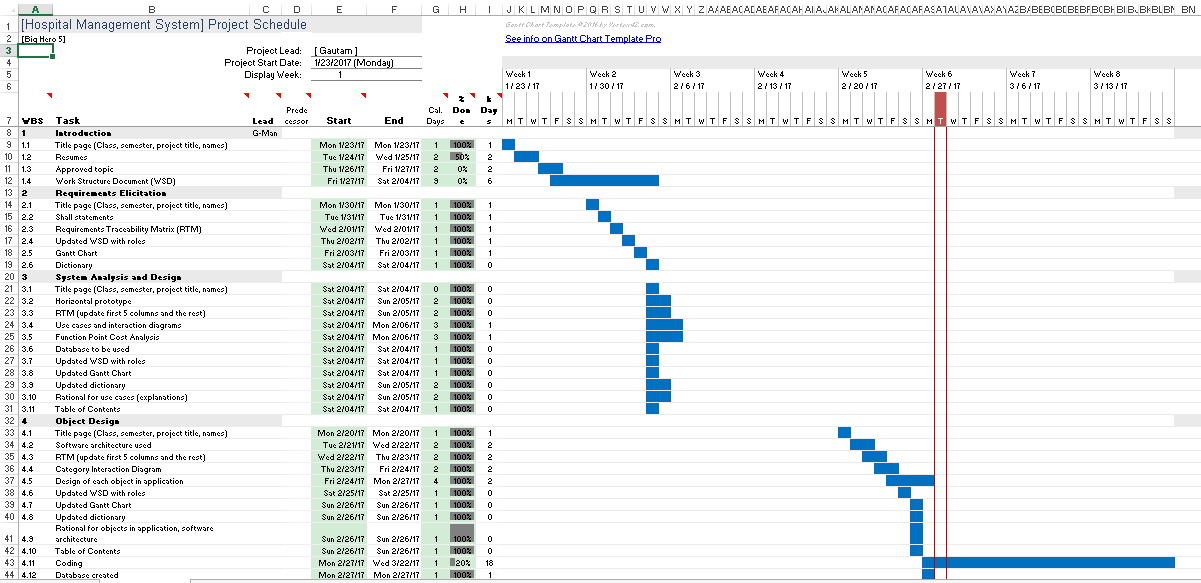
Project Legacy:

This Hospital management system can be used to help make receptionist jobs easier to handle. With just one single input, they are able to get all the information they would need with just a simple ID input along with security measures. This will help the efficiency of hospitals as well as help patients get better service from the hospital. The hospital database system will also allow for better management of doctors so that each patient gets the best service possible in the optimal amount of time. The Pharmacy application will also help patients keep track of their prescriptions as well as help the hospital keep proper stock of their medicine so that every patient will always get what they need immediately.

Final Work Structure Document:

|  |  |
| --- | --- |
| Gautam Ravichandran | **Team Coordinator**  Final Document Handler  Java Coder  Front-end Developer |
| Victoria Green | GUI Coder/tester  Front end tester  RTM Update Manager |
| Ashwin Nair | Java Coder  Front-end Developer  GUI Coder  Final Product Double Checker |
| Karankumar Parikh | Front-end Developer  Database Manager  Setting up Database Backend |
| Young Jun Son | Java Coder  User Guide  Application Tester |

Gantt chart:



Dictionary:

Static:​ This term defines fixed values. These are values that are set to equal a certain value throughout its iteration. This includes the relevant hospital employee information, such as: name, gender, etc. are static as they are fixed values depending on the individual employee information. This also pertains to the same relevant descriptive details of the patients.

Database: ​This is a data storing system which will allow the software to access content of the system later by those including: doctors, nurses, and front desk employees.

GUI:​ (Graphic User Interface) This is the interface in which users will be able to access the content of the Hospital Management System. The GUI contains the UX/UI aspects of the program such as the window, buttons, search bars, tabs, etc.

Java:​ The programming language which is used for the software’s implementation and design. This programming language will address the contents of the Hospital

Management System and the layout of the system methodologies.

Active Data:​ The data that is dynamic or interchangeable within the system. This is the system information pertaining to the employees and the general hospital supply/accommodation and the changes in their data (scheduling/inventory/room availability)

SQLite:​ Database structuring/management tool which stores content related to the user. This user information includes but is not limited to: employee information, patient information, inventory/supply count, general hospital facilities.

UX/UI:​ (User Experience/User Interface) This relates to the content on the front-end side of the software and its accessibility/use or general functionality of the content when used by people for testing purposes or public use.

Use Case Diagram:​ Shows an overview of the system and functions.

Methods:​ These are functions which can be called throughout the program. Sometimes the code for how methods work is hidden from view in order to make code easier to read. These functions complete specific tasks, such as sorting, calculating, and adding/deleting.

Class:​ A class is a blueprint for objects created in a program. These classes will be made to contain organized code to control things like patient data, inventory, and patient accounts.

Object:​ An object is what a class controls. There can be a patient object, which will have states and behaviors such as name, date of birth, and wellness levels.

Horizontal Prototype:​ Shows the broad relationships between a system and maps out its range of abilities.

User Guide:

1.1: Login Screen: Enter username (admin) and password (admin).

1.2: Alternatively, create a user and fill in information (name, date of birth, address, gender, phone number, and type of staff.

2.0: Schedule appointment: Allows user to schedule an appointment by selecting patient and doctor.

2.1: Patient id will allow user to choose patient.

2.2: Patient name and age is displayed.

2.3: Doctor can be chosen.

2.4: Date can be selected, along with time.

2.5: Select reason for visit.

2.6: Click “Schedule” button to confirm.

3.0: Patient info tab.

3.1: Select patient id to bring up information from the database.

3.2: Displays patient name, age, contact number, and upcoming appointments from the database.

4.0: Doctor’s appointments tab shows appointments scheduled for a specified doctor

4.1: Select doctor’s name.

4.2: Appointment schedule is displayed in blue box.

4.3: Contact number for doctor and today’s date is shown.

5.0: Pharmacy tab.

5.1: Low quantity window shows up, warning user. User can now type in a number and order medication. The window will disappear when threshold supply is reached or when the window is manually closed.

5.2: Medicine ID brings up information of medicines.

5.3: Patient ID must be chosen in order to deliver the medicines.

5.4: Quantity to be charged to patient. It will also subtract the quantity from supply.

6.0: Invoice tab shows how much the patient is to be billed.

6.1: Select patient id to show a total amount in dollars, as well as how many medicines were billed.