

Hospital Management Project

Spring Semester 2017

CSC 4350, 5:30-7:15

21 February 2017

Big Hero 5

Gautam Ravichandran

Victoria Green

Ashwin Nair

Karankumar Parikh

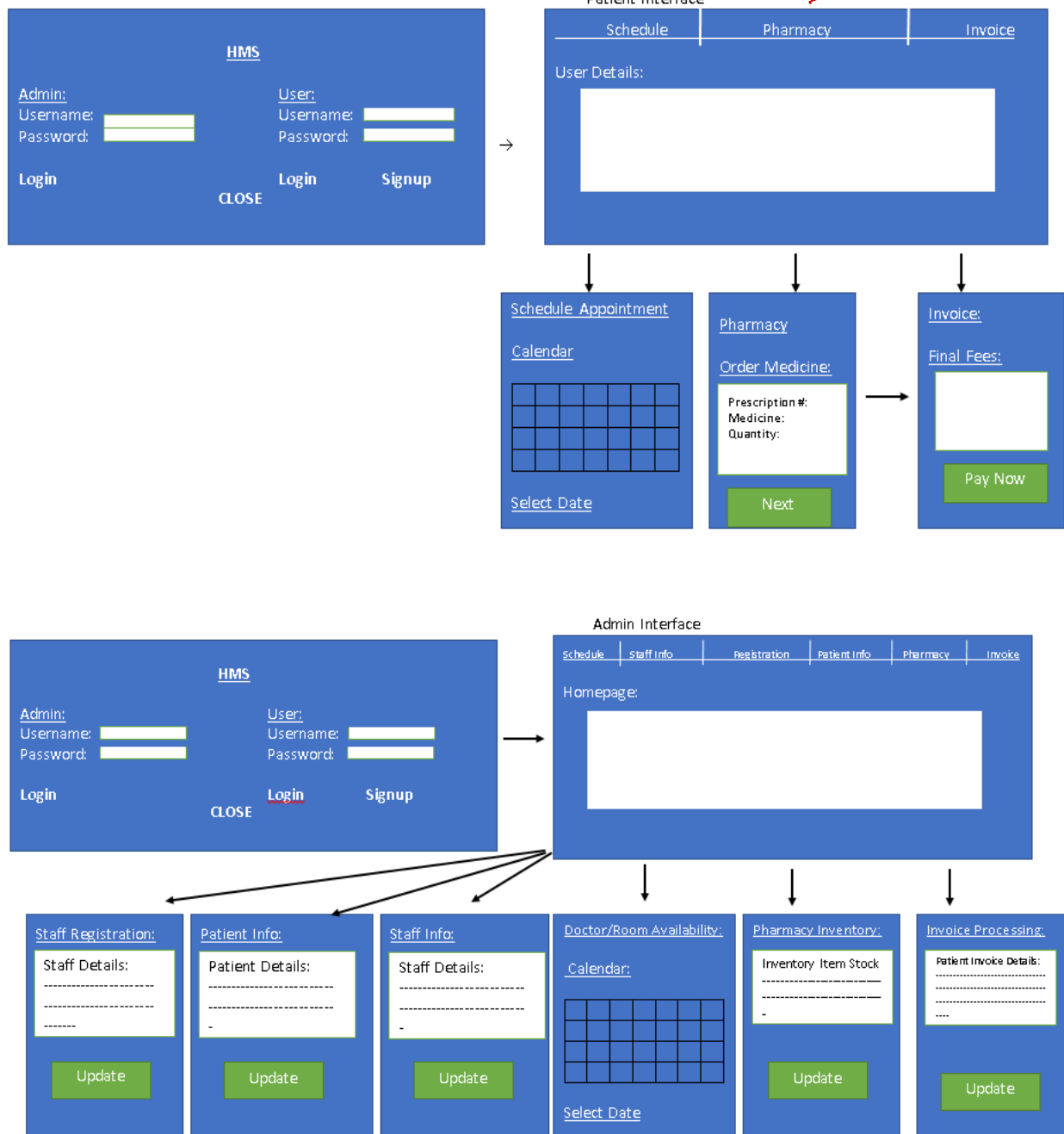
Young Jun Son



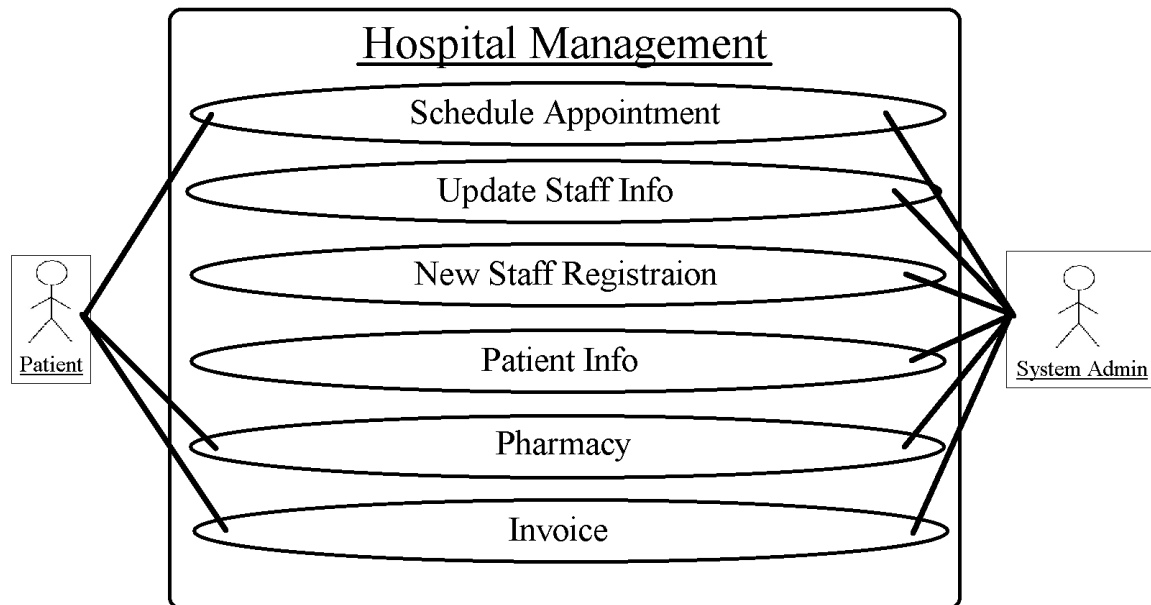
Table of Contents:

Horizontal Prototype	3
Use Case Diagram.....	4
Use Case Rational.....	4
Sequence Diagram.....	5
Required Traceability Matrix.....	6 - 9
Function Point Cost Analysis.....	10
Database.....	11
Work Structure Document.....	11
Gantt Chart.....	12
Dictionary.....	13

Horizontal Prototype:



Use Case Diagram:



Use Case Diagram Rational:

Schedule Appointment allows patients and admin to book an appointment during an available time slot.

Update Staff Info allows the system admin to change list of appointments and list of medications prescribed for all hospital employees as needed. This information should be readily available to patients.

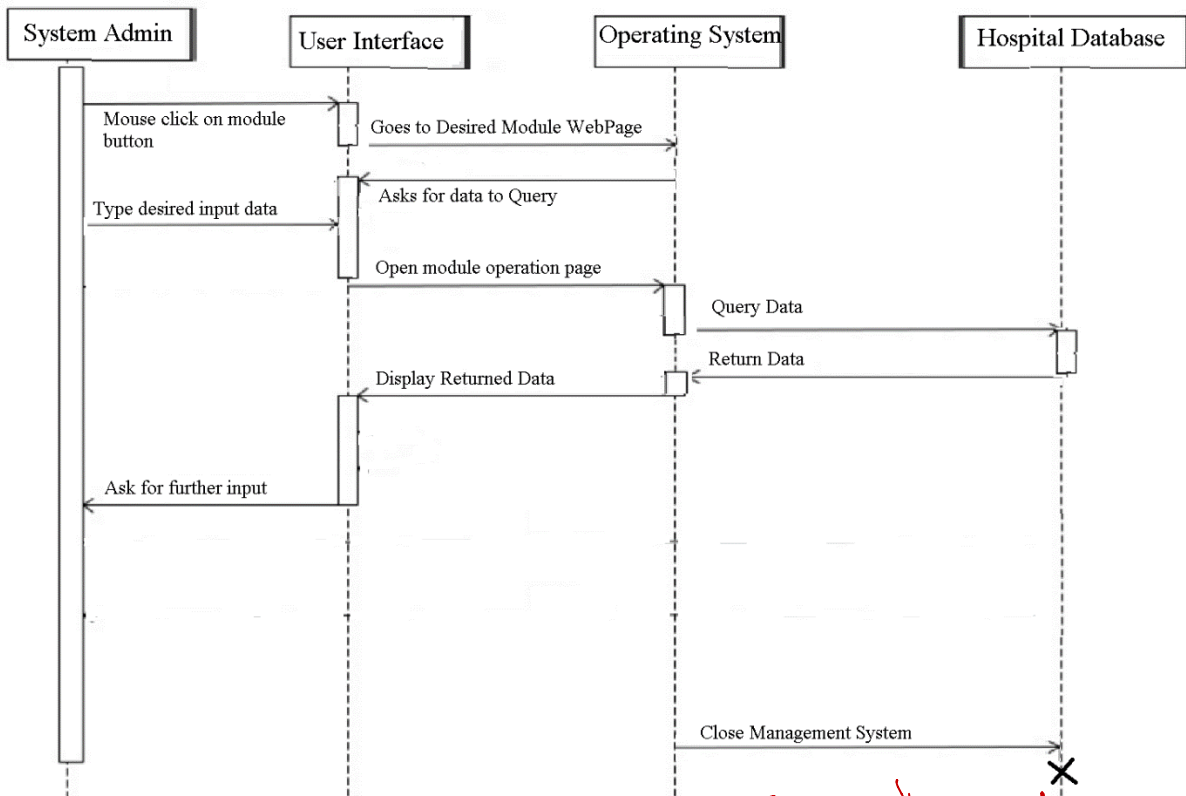
New Staff Registration allows the system admin to input required information for hospital new hires, such as: first and last name, age, gender, place of education, degree level, and year graduated. This information should be readily available to patients.

Patient Info maintains each patient's first and last name, date of birth, reason for visit, illnesses, and any prescribed medications.

Pharmacy maintains the medicine inventory, notifies the system admin when a quantity drops to 3 or under, and distributes prescribed medications to patients.

Invoice is generated after the doctor inputs details from the appointment, and is given to the patient through the system.

Sequence Diagram:



1/ you will need more of these for final document.

Required Traceability Matrix:

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
- Date of Birth
- Reason for Visit

HMS- 3.0 PATIENT RECORDS

3.0.1 The System shall maintain patient records and diagnostic reports 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
- Age
- Gender
- Place of education, degree level, and year graduated

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
- List of medicines prescribed

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

HMS- 5.0 LAB RECORDS

5.0.1 The System shall store lab reports and lab diagnostic information.

5.0.2 The System shall forward the lab reports to assigned doctors.

HMS- 6.0 PHARMACY

6.0.1 The System shall efficiently maintain medicine inventory and notify when a new order needs to be placed.

HMS- 6.1 REORDER SUPPLY

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

- **6.1.1** When the count of an item reaches ≤ 3 , the system shall send a notification to the user with the name of the item and the number on hand.

HMS- 7.0 INVOICE MANAGEMENT

7.0.1 The System shall manage patient invoices and transaction details.

HMS- 7.1 INVOICE CREATION

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

- **7.1.1** When the doctor finishes recording procedures and medicine administered to the patient into the system, an invoice shall be generated.
- **7.1.2** 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, date of birth, reason for visit	SW	B1
3.0.1	3.0	The System shall maintain patient records and diagnostic reports 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, age, gender, place of education, degree level, and year graduated	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, list of medicines prescribed	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 store lab reports and lab diagnostic information.	SW	B2
5.0.2	5.0	The System shall forward the lab reports to assigned doctors.	SW	B2
6.0.1	6.0	The System shall efficiently maintain medicine inventory and notify when a new order needs to be placed	SW	B3

6.1.1	6.1	When the count of an item reaches ≤ 3 , the system shall send a notification to the user with the name of the item and the number on hand.	SW	B3
7.0.1	7.0	The System shall manage patient invoices and transaction details.	SW	B4
7.1.1	7.1	When the doctor finishes recording procedures and medicine administered to the patient into the system, an invoice shall be generated.	SW	B4
7.1.2	7.1	The generated invoice shall add up the cost of everything administered at the appointment.	SW	B4

Function Point Cost Analysis:

FUNCTION - ORIENTED METRICS

Function points [Albrecht 1979] are basic data from which productivity metrics could be computed.

FP data is used in two ways:

- as an estimation variable that is used to "size" each element of the software.
- as baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.

Measurement Parameter	Count	Weighting Factor			
		Simple	Average	Complex	
Number of User Inputs	15 x	<input type="radio"/> 3	<input checked="" type="radio"/> 4	<input type="radio"/> 6	= 60
Number of User Outputs	5 x	<input type="radio"/> 4	<input checked="" type="radio"/> 5	<input type="radio"/> 7	= 25
Number of User Inquiries	4 x	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 6	= 12
Number of Files	3 x	<input checked="" type="radio"/> 7	<input type="radio"/> 10	<input type="radio"/> 15	= 21
Number of External Interfaces	2 x	<input type="radio"/> 5	<input checked="" type="radio"/> 7	<input type="radio"/> 10	= 14
Count = Total -----					132

Note. By clicking on the buttons above more information about the measurement parameters will be available.

0	1	2	3	4	5
No Influence	Incidental	Moderate	Average	Significant	Essential

Rate each factor (Fi, i=1 to 14) on a scale of 0 to 5:

Note. By clicking on the buttons above more information about the measurement parameters will be available.

0	1	2	3	4	5
No Influence	Incidental	Moderate	Average	Significant	Essential

Rate each factor (Fi, i=1 to 14) on a scale of 0 to 5:

F1. Does the system require reliable backup and recovery?	<input type="text" value="4"/>
F2. Are data communications required?	<input type="text" value="4"/>
F3. Are there distributed processing functions?	<input type="text" value="2"/>
F4. Is performance critical?	<input type="text" value="5"/>
F5. Will the system run in a existing, heavily utilized operational environment?	<input type="text" value="3"/>
F6. Does the system require on-line data entry?	<input type="text" value="1"/>
F7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	<input type="text" value="3"/>
F8. Are the master files updated on-line?	<input type="text" value="2"/>
F9. Are the inputs, outputs, files or inquiries complex?	<input type="text" value="4"/>
F10. Is the internal processing complex?	<input type="text" value="3"/>
F11. Is the code designed to be reusable?	<input type="text" value="4"/>
F12. Are conversion and installation included in the design?	<input type="text" value="4"/>
F13. Is the system designed for multiple installations in different organizations?	<input type="text" value="4"/>
F14. Is the application designed to facilitate change and ease of use by the user?	<input type="text" value="4"/>

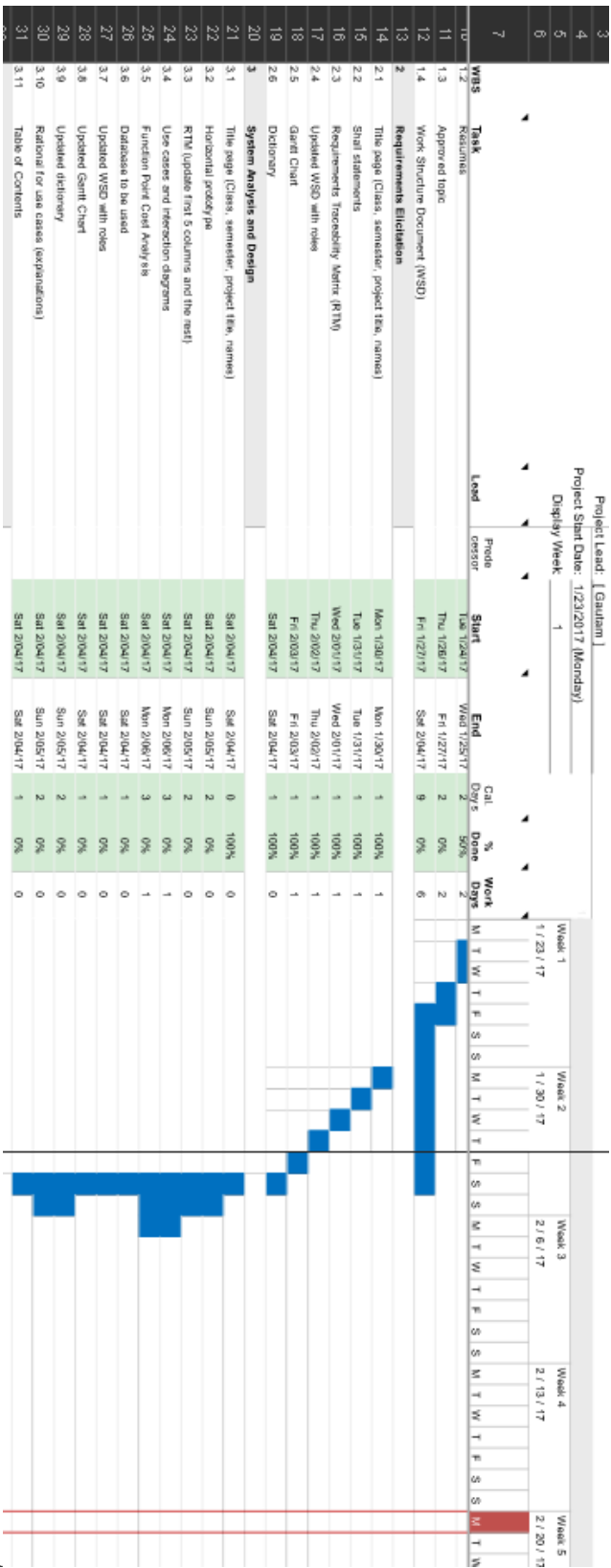
Result. According to the input your project has:

Database: SQLite

Work Structure Document:

Gautam Ravichandran	Team Coordinator Documents Handler Java Coder Front-end Developer
Victoria Green	Database Manager GUI Coder/tester Front end tester
Ashwin Nair	Java Coder Front-end Developer GUI Coder
Karankumar Parikh	Database Manager Setting up Database back end Finalize code documentation
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.