Hospital and Patients Sync Up System

Rel 1.0.0.0

Software Design Document for OO Systems

ARICENT:

Revision Number: 0.1.0.0

Contents

1. 1. Introduction 5

1.1 Purpose 5

1.2 Scope 7

1.3 Definitions and Acronyms 8

1.3.1 Definitions 8

1.3.2 Acronyms 8

1.4 References 8

1.5 Approvals and Authorizations 8

1.6 Distribution 9

1.7 Document Organization 9

1.8 Typographical Conventions 9

1. 2. High Level Design 11

2.1 Architectural Design Description 11

2.1.1 Decomposition Description 11

2.1.1.1 Module Decomposition 11

2.1.1.1.1 Module 1: Login 11

2.1.1.1.2 Module 2: SignUp 11

2.1.1.1.3 Module 3: Patient details 12

2.1.1.1.4 Module 4: Doctor Details 12

2.1.1.1.5 Module 5: Appointment Details 12

2.1.1.1.6 Module 6: Pill Details 12

2.1.1.1.7 Module 7: Notification 12

2.1.1.1.8 Module 8: Manage Staff 13

2.1.1.2 Concurrent Process Decomposition 13

2.1.1.3 Data Decomposition 13

2.1.2 Dependency Description 13

2.1.2.1 Subsystem Dependencies 13

2.1.2.2 Inter-process Dependencies 13

2.1.2.3 Data Dependencies 13

2.1.3 Interface Description 13

2.2 Higher Level Design Description 14

2.2.1 Admin 14

2.2.1.1 addDoctor 14

2.2.1.2 deleteDoctor 14

2.2.2 Patient 15

2.2.2.1 patientDetails 16

2.2.2.2 appointmentDetails 16

2.2.2.2 appointment

2.2.2.4 pillDetails 12

2.2.2.5 viewSpecilizations 12

2.2.2.6 viewDoctorDetails 12

2.2.2.7 notifyAppointment 13

2.2.2.8 pillNotify 13

2.2.3 Specializations 13

2.2.3.1 Appointment 13

2.2.3.2 Cardiology 13

2.2.3.3 Neurology 13

2.2.3.4 Nephrology 13

2.2.3.5 Dermatalogy 13

2.2.3.6 Orthopedics 13

2.2.4 Doctor 17

2.2.4.1 DoctorDetails 17

2.2.4.2 appointmentDetails 17

2.3 Sequence diagrams 18

2.4 Data Flow diagrams 19

2.4.1.1 Level 0 diagram: 19

2.4.1.2 Level 1 diagram: 20

2.4.1.3 Level 2 diagram: 23

2.6 Sequence diagrams 24

2.7 Sequence diagrams 25

CHAPTER 3:

3.1 Alternate Designs 26

3.2 Design Evaluation Report 26

3.3 Design Limitations 26

Template Revision History 35

Figures

Fig 1…………………………………………………………………………………………………………………..13

Fig 2…………………………………………………………………………………………………………………..14

Fig 3…………………………………………………………………………………………………………………..15

Fig 4…………………………………………………………………………………………………………………..16

Fig 5…………………………………………………………………………………………………………………..17

Fig 6…………………………………………………………………………………………………………………..18

Fig 7…………………………………………………………………………………………………………………..19

Fig 8…………………………………………………………………………………………………………………..20

Tables

[Table 1‑1: Definitions Used in this Document 2](#_Toc165179715)

[Table 1‑2: Acronyms Used in this Document 2](#_Toc165179716)

Chapter

1

# Introduction

## Purpose

* Purpose of the software is to provide Patient and Hospital Sync up System for various hospitals. This system will enable patients to book appointments, view their pills and the dosage. This system will also notify patients 1 hour before scheduled pill time and 1 day before the appointment.
* Purpose of the document is to provide high- level design for the system.
* Intended audience of the software is various patients who will register themselves for the product.
* Intended audience of this document is the development team.

## Scope

Name of the software: Aricare

System will do the following things:

1. Enable new patients to register.
2. Enable patients to book appointment.
3. Enable patients to view their pill details;
4. Enable patients to view their appointment history.
5. Automatically send pills and appointment notification to patients.
6. Enable doctors to view their today’s patient appointment details.
7. Enable admin to add and remove doctors, add news and pill details.

## Definitions and Acronyms

### Definitions

Table ‑: Definitions Used in this Document

|  |  |
| --- | --- |
| DFD | It is a diagram which illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.  A **DFD** is often used as a preliminary step to create an overview of the system, which can later be elaborated. |
| SSADM | It is a set of standards developed in the early 1980s for Systems Analysis and Application Design which is widely used for government computing projects in the United Kingdom |
|  |  |

### Acronyms

Table ‑: Acronyms Used in this Document

|  |  |
| --- | --- |
| Acronym | Explanation |
| DFD | Data Flow Diagram |
| SSADM | Structured Systems Analysis And Design Method |
|  |  |

## References

N/A

## Approvals and Authorizations

N/A

## Distribution

Table ‑3: Distribution

|  |  |  |
| --- | --- | --- |
| **Copy No.** | **Holder’s Designation** | **Issue Date** |
| 1 | Naveen Kumar (Mentor) | 28/04/2016 |
| 2 | Aayush Chawla | 28/04/2016 |
| 3 | Keshav Batra | 28/04/2016 |
| 4 | Mandeep Kaur | 28/04/2016 |
| 5 | Sakshi Kansal | 28/04/2016 |
| 6 | Sukanya Sehgal | 28/04/2016 |

## Document Organization

Software Design Document is organized in the following sections:

* Introduction: Purpose of this section is to introduce reader to the system. Explain various terms and acronyms used in the documents and also explain scope of the system.
* High level Design: Purpose of this section is to present high level design of the system via DFDs, Architectural design descriptions and class descriptions.
* Alternate Design: Purpose of this section is to present alternative design of the system to the reader using its scope, requirements and advantages and disadvantages.
* Low Level Design: Purpose of this section is to give the internal logical design of the actual program code.

## Typographical Conventions

* The convention used in this document is **SSADM** (for DFDs).

Chapter

2

# High Level Design

## Architectural Design Description

### Decomposition Description

#### Module Decomposition

Following are the modules which are available in our system.

##### Module 1: Login

* This module will allow admin, patients and doctors to login in the application
* The details required are Contact number, Password and Role.
* The data provided will be validated from the database.
* This module has no dependencies on other modules.

##### Module 2: Sign Up

* This module will register patients on the application.
* The user details required are name,contact number,address,email id and password..
* This data will be saved in the patient\_details database.
* This module has no dependencies on other modules

##### Module 3: Patient details

* This module stores details of the patients in the database. The patients can view their profile,medical details and appointment history and request for the appointment ,.
* The user details required is: Contact Number and password.
* This module has dependencies on login module.

##### Module 4: Doctor Details

* This module will be used to store details of all the doctors..
* List of doctors will be displayed for that particular specialization, patient

can select a particular doctor name and can see the doctor’s details.

* This data will be fetched from doctor database.
* This module has dependencies on login module.

##### Module 5: Appointment Details

* This module will allow patients to directly schedule an appointment with a doctor or select doctor from specialization and make appointment.
* This data will be saved in the database.
* This module will allow doctors to view their appointment schedule.
* This module has dependencies on patient and doctor modules.

##### 2.1.1.1.6 Module 6: Pill Details

* This module will be used to store the details of all the pills prescribed to the patients.
* The user details required is patient’s contact number and password.
* This data will be saved in the pill database and retrieved when required.
* This module has dependencies on patient module and doctor module.

##### 2.1.1.1.7 Module 7: Notification

* This module will be used to notify the patients 1 day before their scheduled appointment and 1 hour before their prescribed pill time.
* This module has dependencies on Pill details and Appointment details module.

##### 

##### 2.1.1.1.8 Module 8: Admin Module

* This module will be used to add or delete doctors from the system.
* The details required to add doctor are:name,qualification,specialization, address and contact details.
* This data will be saved in the doctor database.
* The details required to delete doctor is:contact details.
* This module has no dependencies on other modules

#### Concurrent Process Decomposition

N/A

#### Data Decomposition

N/A.

### Dependency Description

#### Subsystem Dependencies

N/A.

#### Inter-process Dependencies

N/A.

#### Data Dependencies

N/A

### Interface Description

N/A

## Higher Level Design Description

### Admin

Following classes are possible as per our design:

#### addDoctor

It will add new Doctor to the Database with its complete profile information.

.

#### deleteDoctor

Doctor details will be deleted from the database and its corresponding appointment with patient will be also deleted.

#### updatePillDetails

Patient pill details(start date,end date,timing and dosage) will be updated in the

Database.

#### addNews

Admin will write subject and content of newsletter which will be sent to all the registered patients.

### Patient

### 

#### patientProfile

Patient can view their profile.

#### appointmentHistory

Patient will be able to see his/her appointment history including

Appointment date and time..

#### appointment

Patient will be able to book his/her appointment for a particular doctor and available time slots.

#### pillDetails

Patient will be able to see his/her pill details including start date,end date,

timing and dosage.

#### viewSpecializations

Patients will be able to view all the list of doctors based on their

Specialization.

#### viewDoctorDetails

Patients will be able to view all the details of the doctors of the hospital.

#### notifyAppointment

Notification pop will be displayed to patient 1 day before the appointment .

#### pillNotify

Patient will be notified one hour before intake of pills.

### Specialization

#### Appointment

Appointment for patient will be fixed according to specialization.

#### Cardiology

It will fetch the list of cardiology doctors from the database.

#### Neurology

It will fetch the list of Neurology doctors from the database.

#### Nephrology

It will fetch the list of Nephrology doctors from the database.

#### Dermatology

It will fetch the list of Dermatology doctors from the database.

#### Orthopedics

It will fetch the list of Orthopedics doctors from the database.

### Doctor

#### doctorProfile

Doctor can view their details.

#### appointmentDetails

Doctor can see all his appointments and the details will be fetched from database.

## Data Flow diagrams

#### Level 0 diagram:

C:\Users\gur44608\Downloads\dfd(0).png

Fig 1

#### Level 1 diagram:

C:\Users\gur44629\Downloads\dfd level1 (1).png

Fig 2

C:\Users\gur44629\Downloads\doctor dfd level1.png

Fig 3

C:\Users\gur44629\Downloads\admin(1).png

Fig 4

#### Level 2 diagram:

C:\Users\gur44629\Downloads\dfd2.png

Fig 5

## 2.5 Sequence diagrams

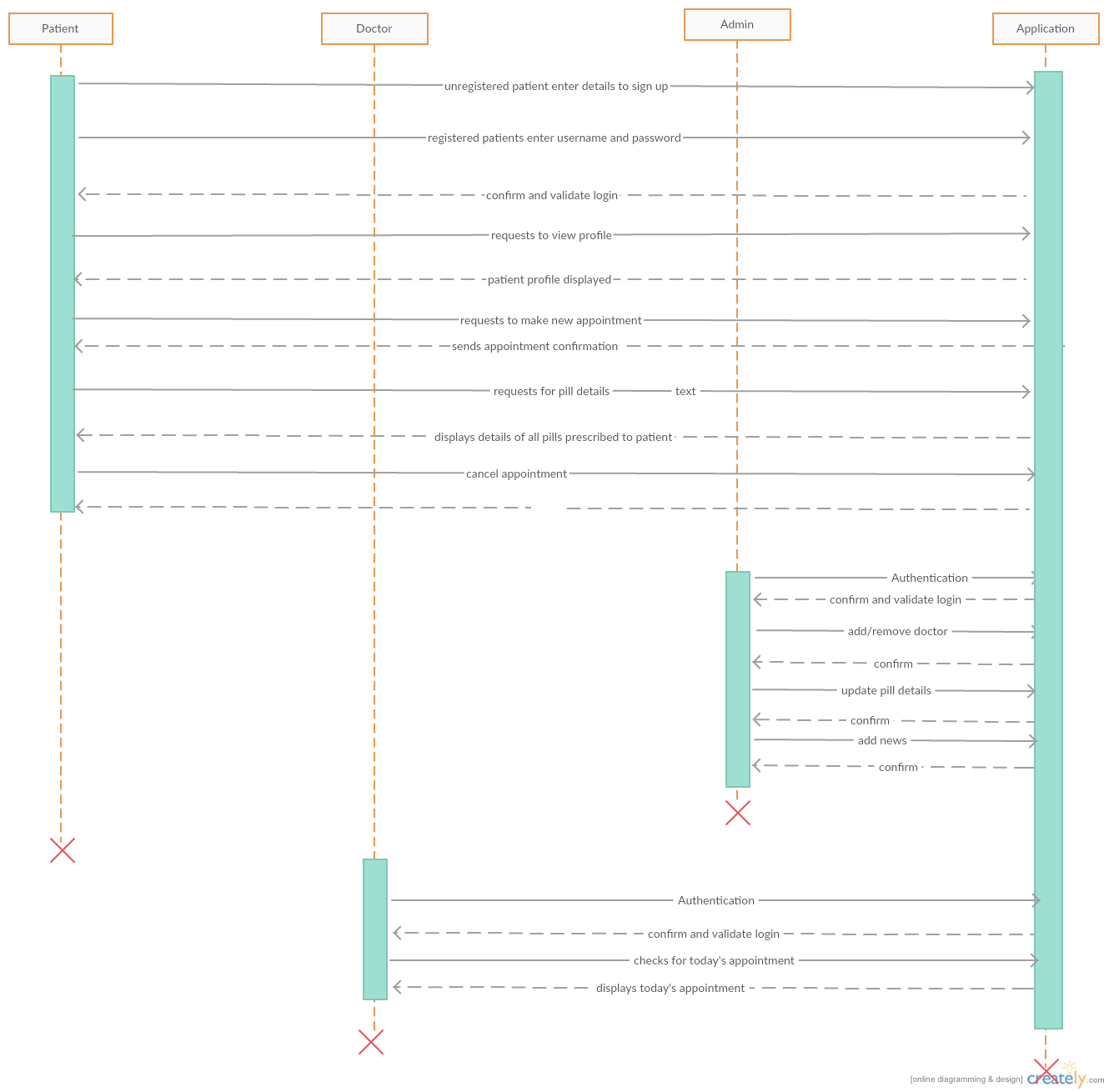


Fig 6

## 2.6 Flowchart

C:\Users\gur44629\Downloads\flow.png

Fig 7

## 2.7 ER Diagram

+

C:\Users\gur44608\Downloads\erd (2).png

Chapter

3



## Alternate Designs

* The alternate design could be a console based application.
* This design will be a desktop application that can be used

## Design Evaluation Report

Advantages of this alternate design:

* System will be simple.
* Direct patients can be entertained if the slots are free.
* Flexible.

Disadvantages:

* User cannot select a time according to choice and priorities.
* System not accessible everywhere.

## Design Limitations

* Late patients are not entertained.
* Time slots are fixed.

Appendix

###### Design Entity Attributes

Identification: The name of the entity should be specified. Two entities should not have the same name. The names for the entities may be selected to characterize their nature.

Type: The type attribute should describe the nature of the entity. It may simply name the kind of entity, such as subprogram, module, procedure, process, data item, object etc. Alternatively, design entities can be grouped, in order to assist in locating an entity dealing with a particular type of information.

Purpose: This is a description of why the entity exists. It provides the rationale for the creation of the entity. Therefore it designates the specific functional and performance requirements for which this entity was created, using the SRS.

Function: A statement of what entity does. The function attribute should state the transformation applied by the entity inputs to produce the desired output. In the case of a data entity, this attribute should state the type of information stored or transmitted by the entity.

Subordinates: the identification of all entities composing this entity. The subordinate attribute should identify the entities composing this entity. This information is used to trace requirements to design entities and to identify the parent/child structural relationships through software system decomposition.

Dependencies: the description of the relationships of this entity with other entities. The dependencies attribute should identify the relationship of the entity with other entities. It describes the nature of each interaction that may involve initiation, order of execution, data sharing, creation, duplicating, usage, storage or destruction of other entities.

These relationships are depicted by state charts, data flow diagrams, transaction diagrams.

Interface: A description of how other entities interact with this entity. The interface attribute describes how other entities interact with this entity. It should describe the methods of interaction and rules governing those interactions. It provides a description of the input ranges, the meaning of inputs and outputs, the type and format of each input or output, and output error codes.

Resources: A description of elements used by entity that are external to the design. The resources attribute identifies and describes all of the resources external to the design that are needed by this entity to perform its function. It provides information about items such as physical devices (printers, discs, memory), software services (math libraries, operating system services, graphical user interface libraries), and processing resources (CPU cycles, memory allocation).

Processing: A description of the rules used by the entity to achieve its function. The processing attribute describes the rules used by the entity to achieve its function. It describes the algorithm used by the entity to perform a specific task. It includes sequencing of events or processes, process steps, conditions, termination criteria etc. Describe the control and data flow between immediate sub design entities of this design entity. If the design entity has no immediate sub design entities (i.e. it is fully decomposed) then outline the method of processing used by the design entity to perform its task (e.g. with pseudo-code, state diagrams, etc).

Data: A description of data elements internal to entity. The data attribute describes the method of representation, initial value, use, format and acceptable values of internal data. Describe in detail (where possible) the local data values and data structures belonging to (local in scope) this entity. Otherwise give an outline description.

Template Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rev | Date of Issue | Author | Approver | Scope |
| 1 | 28/04/2016 | TH3\_GR03 | Naveen Kumar |  |
|  |  |  |  |  |
|  |  |  |  |  |