

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018



Group 10

Master Clinic/10

Project Master Clinic SDS 1.0CM_Id 04,01,2018
Master Clinic Statement of Work1.0_mc_04 01, 2018

Master Clinic

Software Design Specification (SDS)

Team 10 Master Clinic 1.1

IdDateVersion: 1.0

CM Identifier: Master Clinic1.0_mc_04 01, 2018

<The Software Design Specification (SDS) sections given in this outline are guidelines to the contents of your SDS. Include *at least* these sections. Your team may have good reasons for wanting to deviate from this proposed outline. In this case, you *must* motivate any deviations to me. If a section is not applicable in your case, do not delete it; instead, give the topic heading and write "Not applicable".

You will note that there is some overlap in the contents from document to document. This redundancy allows each document to stand on its own. >

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

Revision History

Sl. No.	Prepared/ Modified by	E-mail	Version	Date	Approved by	Descriptions/ Remarks
1.	Abdelrahman Mohamed	am535150@gmail.com	<1.0>	4/9/2018	Hady Maher	System Architecture/ Design Models/ System Deployment.
2.	Aisha Mousa	aishamousa632@yahoo.com	<1.0>	4/9/2018		
3.	Hady Maher	hadymaher311@gmail.com	<1.0>	4/9/2018		
4.	Reham Hamdy	rehamhy6@gmail.com	<1.0>	4/9/2018	Aisha Mousa	Introduction/ Data Models/ Traceability to requirements.

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

Distribution list

Name	E-mail	Notes
Dr: Ahmed Hamdy		
TA: Ali El-Sedeek	alielseddeek@gmail.com	
TA: Dina El Reedy	dinaelreedy@gmail.com	

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

Table of Contents

Introduction	6
Purpose of this Document	6
Scope	6
Table of Acronyms and Definitions	7
Definition	7
References	7
Overview of Document	7
System Architecture	8
Relation between controllers and models	8
Relation between views and controllers	8
Admin	8
Nurse	8
Patient	8
Guest	9
Design Models	9
Design Patterns Description	9
MVC Architectural design pattern	9
Class Diagrams	9
User	10
properties	10
Methods	10
Admin	10
properties	10
Methods	10
Nurse	11
properties	11
Methods	11
Patient	11
properties	12
Methods	12
Clinic	12
properties	12
Methods	12

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

Model	12
Methods	12
Interaction Diagrams	14
Data Models	30
System Deployment	32
Client tier	33
Server tier	33
Traceability to Requirements	33

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

List of Tables

Table-1 :Traceability to Requirements	23
--	-----------

List of Figures

Figure-1 : System Architecture	9
Figure-2 : Class Diagram	13
Figure-3 : SD1	14
Figure-4 : SD2	15
Figure-5 : SD3	16
Figure-6 : SD4	17
Figure-7 : SD5	18
Figure-8 : SD6	19
Figure-9 : SD7	20
Figure-10 : SD8	21
Figure-11 : SD9	22
Figure-12 : SD10	23
Figure-13 : SD11	24
Figure-14 : SD12	25
Figure-15 : SD13	26
Figure-16 : SD14	26
Figure-17 : SD15	28
Figure-18 : SD16	28
Figure-19: SD17	29
Figure-20: SD18	29
Figure-21 : er- entities	30
Figure-22 : er- relations	31
Figure-23 : er- map	31
Figure-24 :Deployment-diagram	33

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

1. Introduction

Dental care is very important as teeth affects both health and appearance. However lots of people go to the dentist with an average ratio of 64% of a yearly visit for adults. Managing patients' paper and registering a dentist becomes really an annoying task. That's why we are creating **MASTER-CLINIC** which is a website that hopefully would help you register a dentist, manage your treatment paper and expenses and also help dentists in managing their patients' files and easily getting statistics and finding specific patient's data at any time.

1.1 Purpose of this Document

The purpose of this document is to provide an overview of the design specifications employed to implement the requirements stated in the SRS document .The software design of master-clinic includes the system architecture which describes the different relations between system components e.g. controllers and views. It also includes design models such as design patterns and diagrams illustrating the design elements and their interactions.The design uses a data model (er-diagram) and explains the System deployment plan .

1.2 Scope

The project is mainly concerned with building a centralized database system for a master-clinic business, and building modules that allows three types of users (patients-nurses-doctors) to interact together in a fast and easy way. patients can make reservations with their doctors,check their medical file through a web application. Nurses confirm patients' reservations,update patients' information, delete patients ,create invoices and manage clinics. Doctors are concerned with patients' files , in terms of their creation ,update and deletion , alongside other administrative functions nurses enjoy like accessing patient's information and clinic management,doctors can also access nurses' information and perform all sort of data manipulation.

1.3 Table of Acronyms and Definitions

Term	1.3.1 <i>Definition</i>
Worker	anyone working in the clinic and not a doctor or nurse.
admin	in this project admin is the dentist himself
Super admin	clinic owners
Patient file	a file that contains all patient's prescriptions, photos and any other comments or data dentist noted about the patient like case description and progress.

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

1.4 References

1.5 Overview of Document

- Section 2 : description of the system architecture.
- Section 3 : layout of design models used in the system e.g. design patterns, class diagrams...etc
- Section 4 : data models of the system illustrated by an er-diagram.
- Section 5 : explaining the deployment plan of the system in terms of tiers.
- Section 6 : a table to trace requirements to their corresponding design elements.

2. System Architecture

Since the project is website then it would be used by different users at the same time. This means that a single point of control won't be suitable. So we use MVC architecture which consists of three separate layers. Views layer which is the GUI that takes inputs and actions from users. Controllers layer which are the processing layer in which the application logic is running and have more than one controller that are able to function all together at the same time. Finally the Models layer which is the data layer that contains the application data and connects with the database. MVC architecture makes relation between different layers like the following views are connected to controllers and controllers are connected to models. We start with relation between controllers and models.

Relation between controllers and models

- Admin Controllers use all the models since admins are allowed to create, update or delete any other entity and themselves given the right permissions (only super admins can delete or update any other admins and they are the only ones who can create other super admins. Other than that there is no other restriction for normal admins).
- Nurse Controllers use both Nurse model and Patient models since a nurse can update herself and create, update or delete a patient
- Patient Controllers use only Patient model as a patient can only update himself

Relation between views and controllers

Since the navigation bar contains a link for logging out (If you are logged in if not it's replaced with a link to the login page) and a link for the home page this makes them every user (admin, nurse, patient or guest) views capable of accessing both the home controller and login controller. That's why no connections to these 2 controllers from their respective views would be mentioned after in the next section.

Admin

- login page: can only access forgot password controller
- send reset email page: can only access forgot password controller
- reset password page: can only access reset password controller
- rest of the views: can access patient, nurse, admin and profile controllers due to the links that includes all these views in the side navigation bar. These links would be available in every single view as soon as admin is logged in to make it easier to perform different functionalities

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

3. Design Models

3.1 Design Patterns Description

MVC architecture which consists of three separate layers. views layer which is the GUI that takes inputs and actions from users. Controllers layer which are the processing layer in which the application logic is running and have more than one controller that are able to function all together at the same time. Finally the Models layer which is the data layer that contains the application data and connects with the database. MVC architecture makes relation between different layers like the following views are connected to controllers and controllers are connected to models.

3.1.1 MVC Architectural design pattern

3.2 Class Diagrams

Class diagram is meant to show system different classes, their attributes and operation (methods). So we have five classes Patient, Admin, Nurse, User and Clinic. There is also interfaces which are some relevant functions packed together in one structure and we have one interface Model which contains functions to interact with database. We start with class admin

User

User is an abstract class that handles system users with basic data and functions.

properties

User class have not properties as it's only to handle basic users functions. Every user would implement his own properties

Methods

1. login: this methods is for handling user login it takes user's email and password as arguments to check if the user already exist and authorize his access

Admin

Admin class is a user class that inherits User class and represents admin user.

properties

1. id: integer that represents admin id number which is used to identify the admin thus no two admins would have the same id
2. name: string that represents admin's name (usually first name)
3. email: string that represents admin's email address and must be unique as it's used for logging into the system
4. password: string that represents admin's password and is used to authorize logging in
5. image: string represents path admin's profile picture in the storage files
6. mobile: string that represents admin's mobile number
7. status: boolean variable that indicates account status inactive (false) and active (true)

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

8. role: string that represents the role of the admin with two valid roles normal and super

Methods

1. createPatient: methods for creating patients which takes an array of patient's data create the patient if the data is valid and then return the new patient
2. updatePatient: methods for updating patients which takes an array of patient's data update the patient if the data is valid
3. deletePatient: method for deleting patients that takes the patient argument then delete him
4. listPatients: methods that returns a list of all available patients
5. createNurse: methods for creating nurses which takes an array of nurse's data create the nurse if the data is valid and then return the new nurse
6. updateNurse: methods for updating nurses which takes an array of nurse's data update the nurse if the data is valid
7. deleteNurse: method for deleting nurses that takes the nurse as argument then delete her
8. listNurses: methods that returns a list of all available nurses
9. createClinic: methods for creating clinics which takes an array of clinic's data create the clinic if the data is valid and then return the new clinic
10. updateClinic: methods for updating clinics which takes an array of clinic's data update the clinic if the data is valid
11. deleteClinic: method for deleting clinics that takes the clinic as argument then delete him
12. listClinics: methods that returns a list of all available clinics
13. createAdmin: methods for creating admins which takes an array of admin's data and string role (to verify permissions normal admin can create normal admin and super admin can create both super and normal admins) create the admin if the data is valid and then return the new admin
14. updateAdmin: methods for updating admins which takes an array of admin's data and role (only super admin can update other admins) update the patient if the data is valid
15. deleteAdmin: method for deleting admins that takes the admin and role (only super admin can delete other admins) as argument then delete him
16. listAdmins: methods that takes role (only super admin can list all admins) as an argument returns a list of all available admins

Nurse

Nurse class is a user class that inherits User class and represents nurse user.

properties

1. id: integer that represents nurse id number which is used to identify the nurse thus no two nurses would have the same id
2. name: string that represents nurse's name (usually first name)
3. email: string that represents nurse's email address and must be unique as it's used for logging into the system
4. password: string that represents nurse's password and is used to authorize logging in
5. image: string represents path nurse's profile picture in the storage files
6. mobile: string that represents nurse's mobile number
7. status: boolean variable that indicates account status inactive (false) and active (true)
8. gender: nurse gender (male/female)

Master Clinic	CM-identifier: MC SE02 v1.0
Software Design Specification	Date: 01/04/2018

9. dateOfBirth: nurse date of birth
10. salary: nurse monthly salary
11. clinicId: id of the clinic is working in

Methods

1. createPatient: methods for creating patients which takes an array of patient's data create the patient if the data is valid and then return the new patient
2. updatePatient: methods for updating patients which takes an array of patient's data update the patient if the data is valid
3. deletePatient: method for deleting patients that takes the patient argument then delete him
4. listPatients: methods that returns a list of all available patients

Patient

Patient class is a user class that inherits User class and represents patient user.

properties

1. id: integer that represents nurse id number which is used to identify the nurse thus no two nurses would have the same id
2. name: string that represents nurse's name (usually first name)
3. email: string that represents nurse's email address and must be unique as it's used for logging into the system
4. password: string that represents nurse's password and is used to authorize logging in
5. image: string represents path nurse's profile picture in the storage files
6. mobile: string that represents nurse's mobile number
7. status: boolean variable that indicates account status inactive (false) and active (true)
8. gender: patient gender (male/female)
9. dateOfBirth: patient date of birth

Methods

Patient class can only update himself, login and logout which are all inherited from class User so there is methods added by this classin

Clinic

Clinic class represents clinics which unlike previous classes is not a user. That's why it doesn't inherit User class

properties

1. id: integer that represents clinic id number which is used to identify the clinic thus no two clinics would have the same id
2. name: string that represents clinic's name
3. email: string that represents clinic's email address which is used for contacting with patients
4. telephone: string that represents clinic's telephone number
5. openingTime: the time the clinic opens
6. closingTime: the time the clinic closes

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

Methods

Unlike the previous classes this is not a user clinic class so it doesn't have any methods at all.

Model

This is an interface used to interact with database. It provide functions to make secure connections with database and perform different operation on it.

Methods

1. find: method that gets row by id from its table
2. update: method that takes an array of data and update a row in database table
3. insert: method that takes an array of data and creates a new row in a database table
4. delete: method that deletes a row by its id

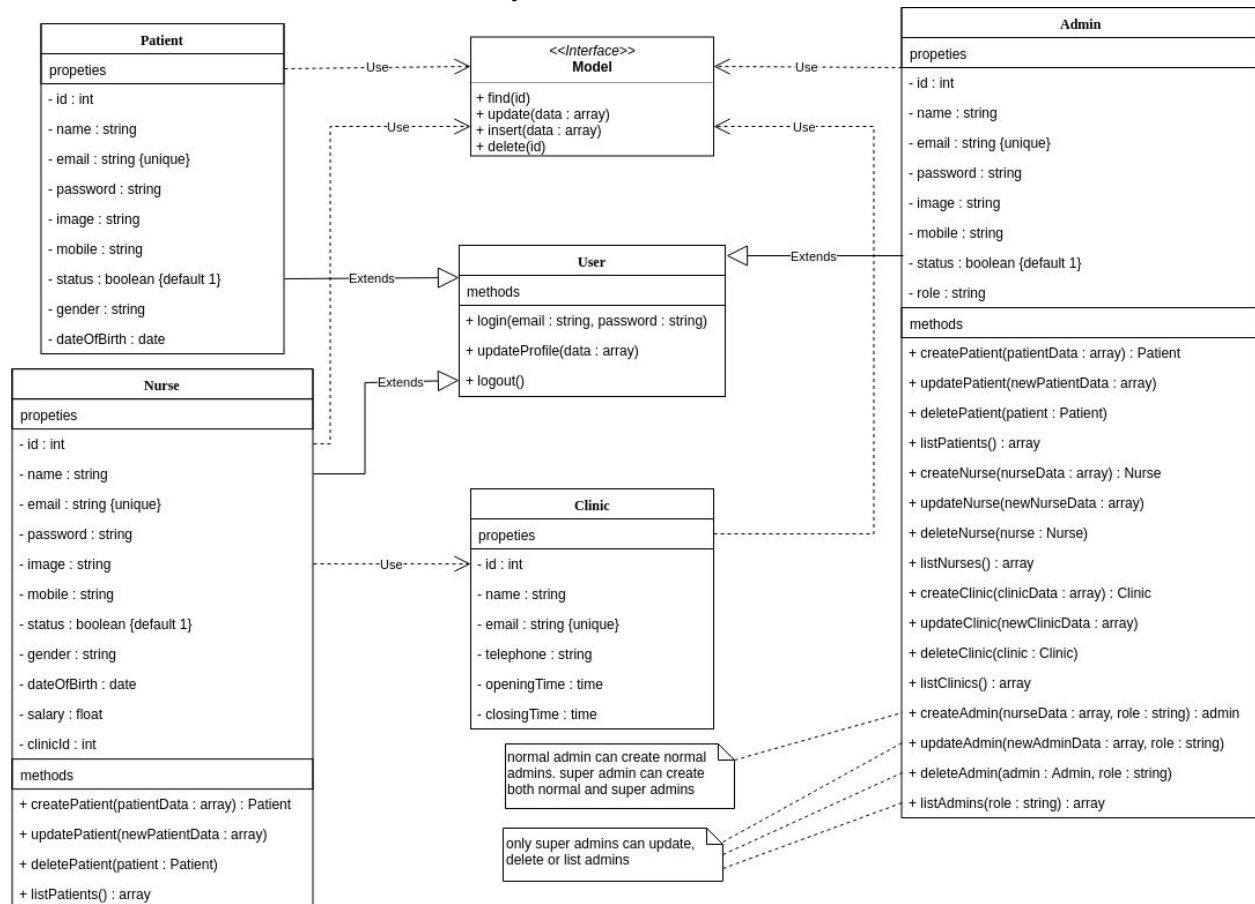


Figure 2: Class diagram

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

3.3 Interaction Diagrams

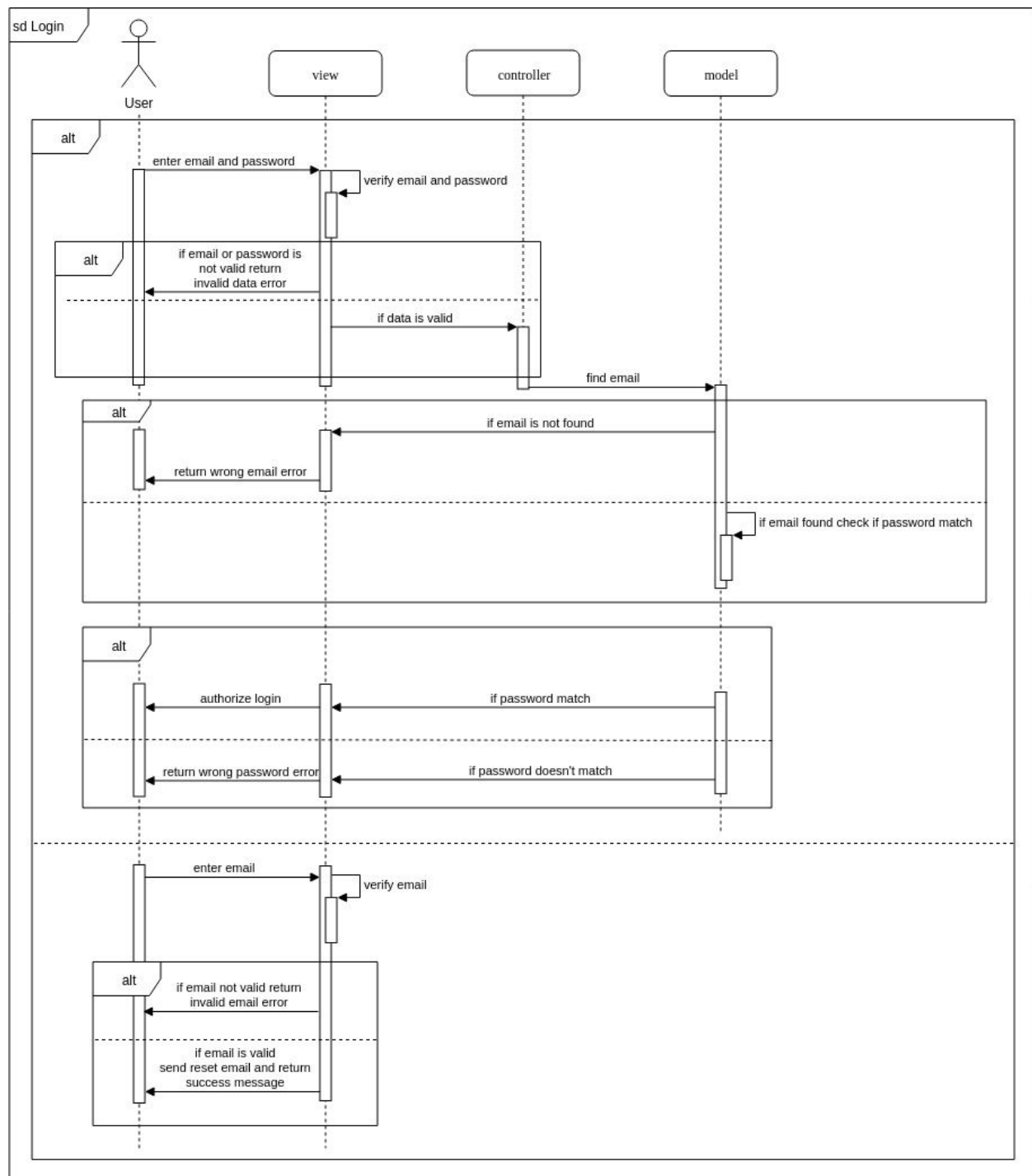


Figure 3: SD1

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

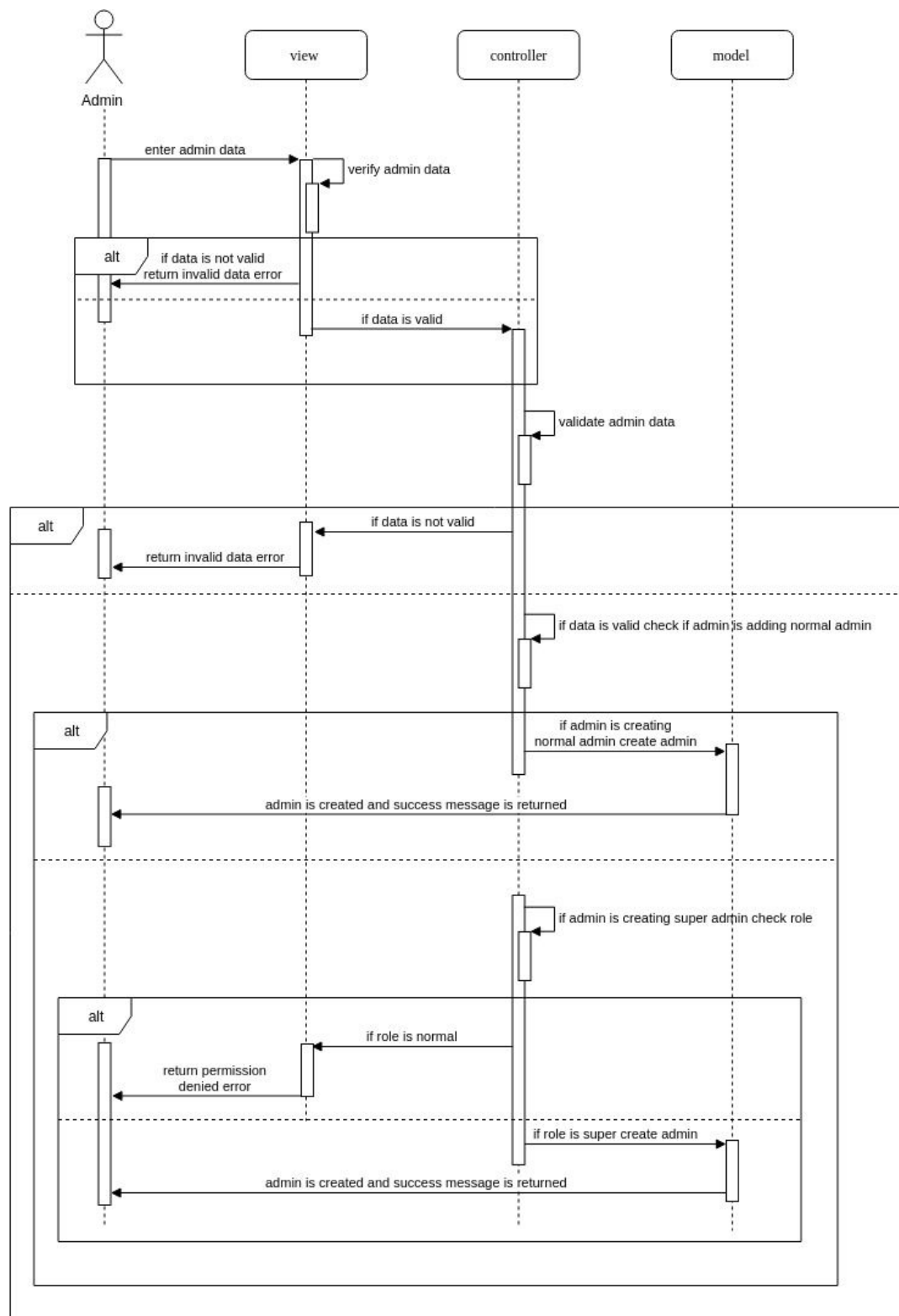


Figure 4: SD2

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

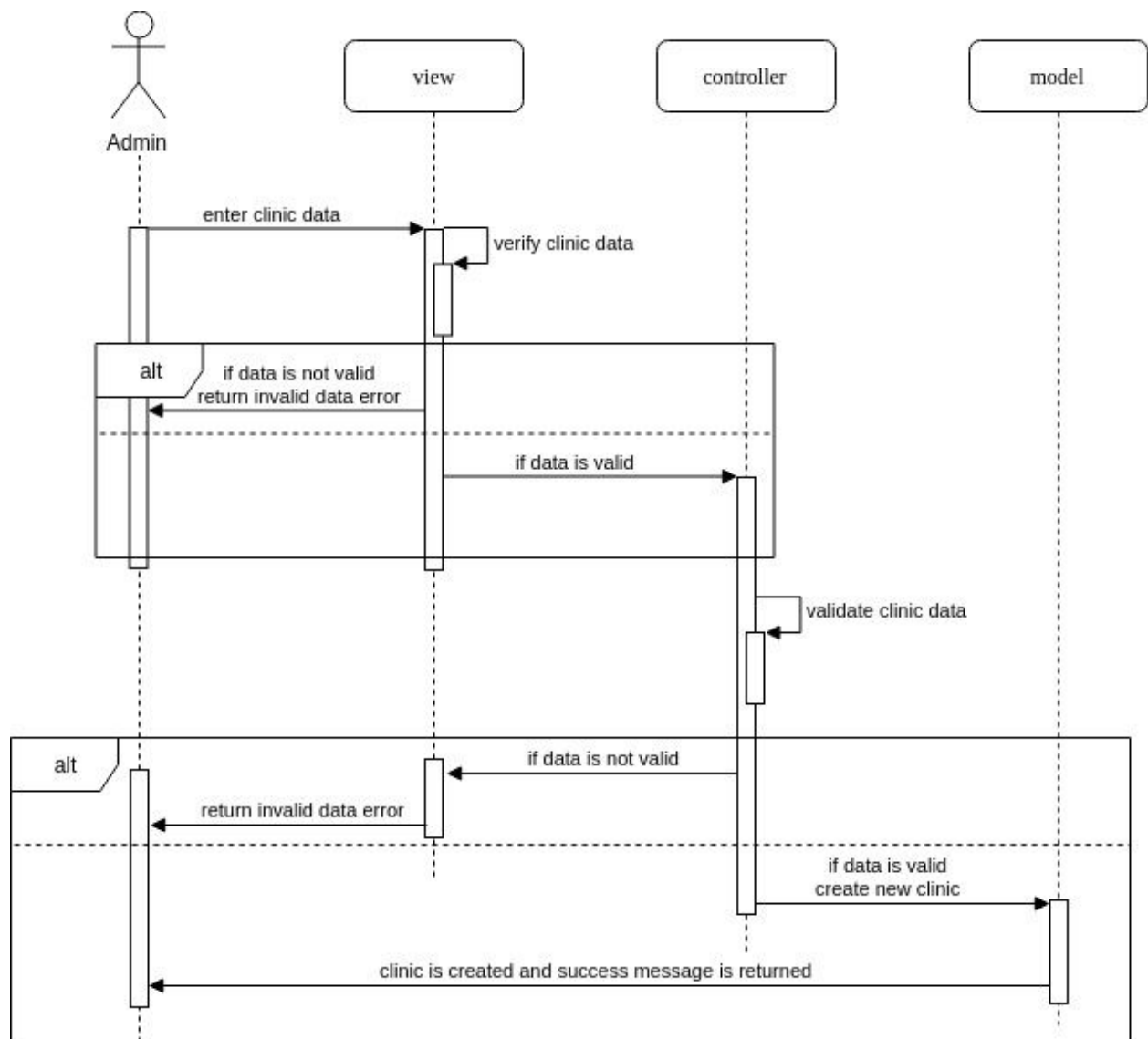


Figure 5: SD3

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

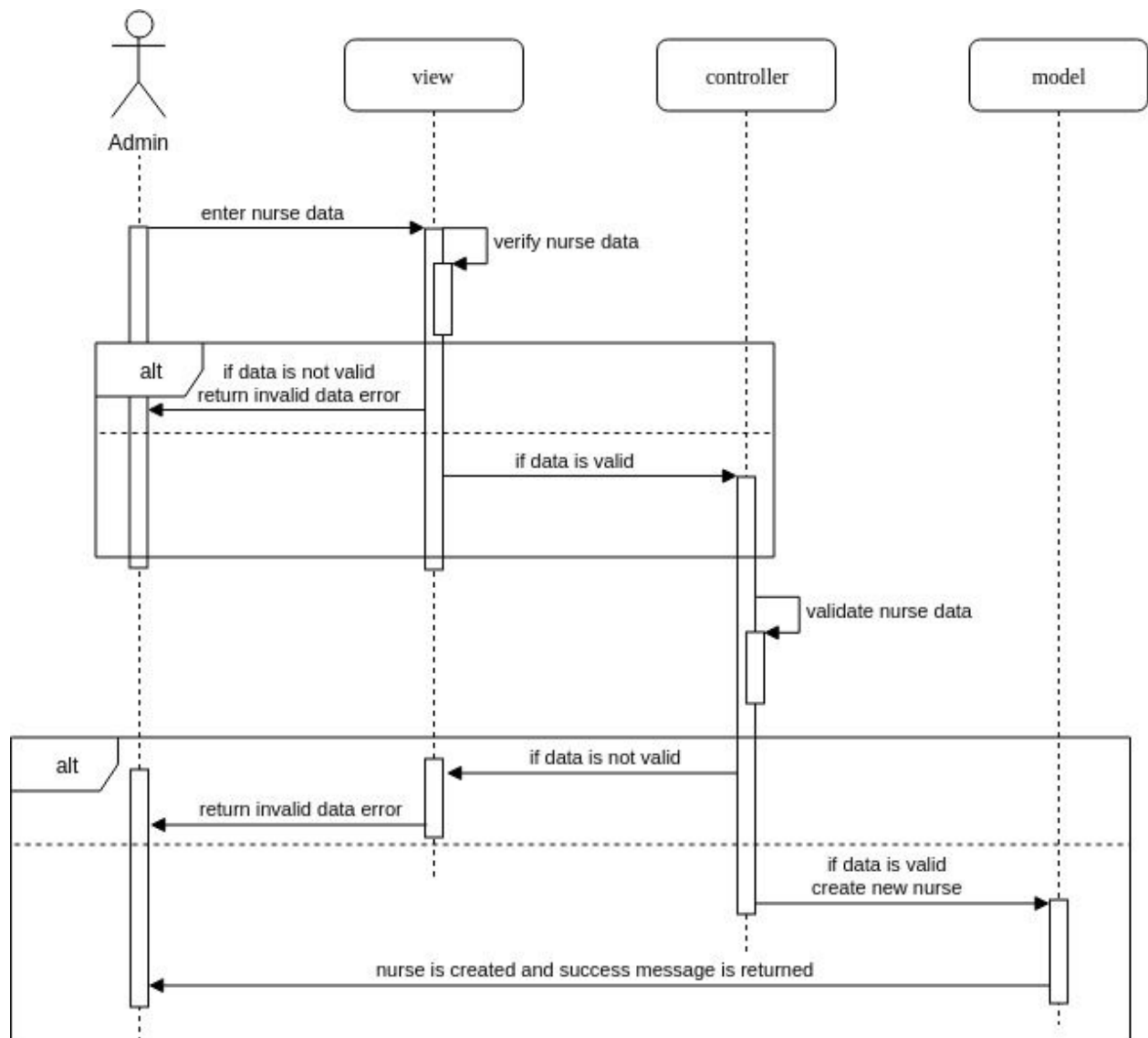


Figure 6: SD4

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

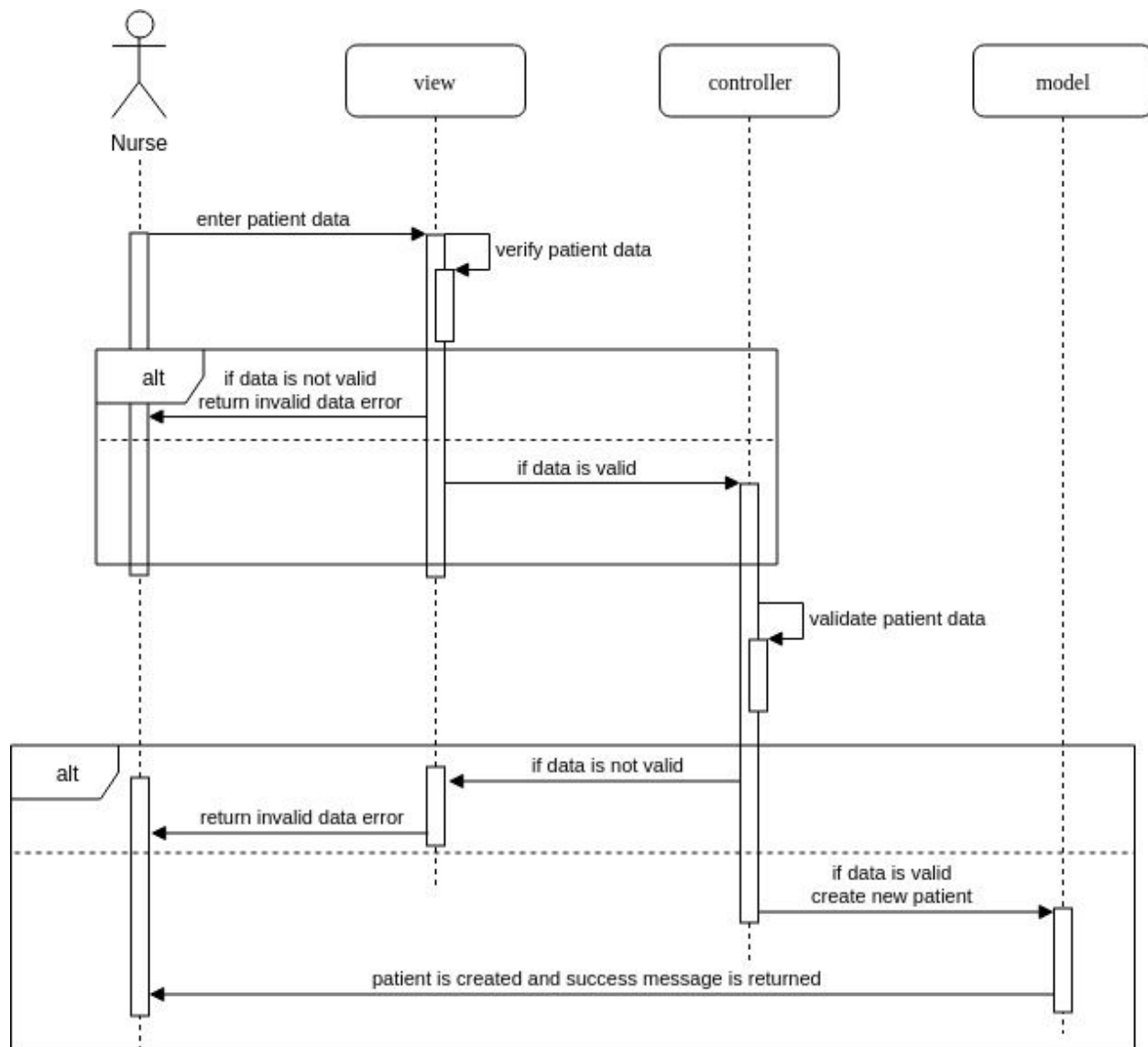


Figure 7: SD5

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

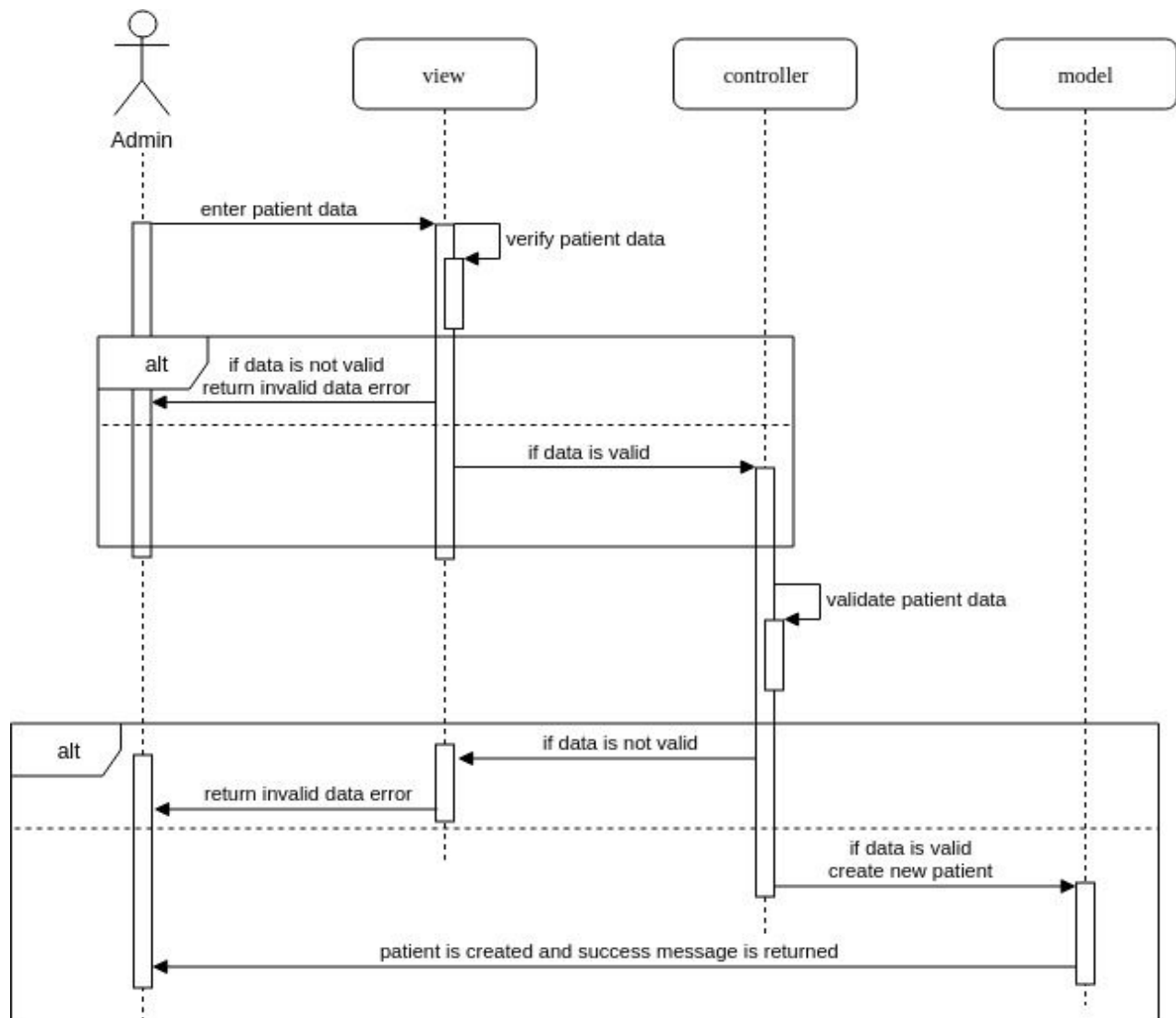


Figure 8: SD6

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

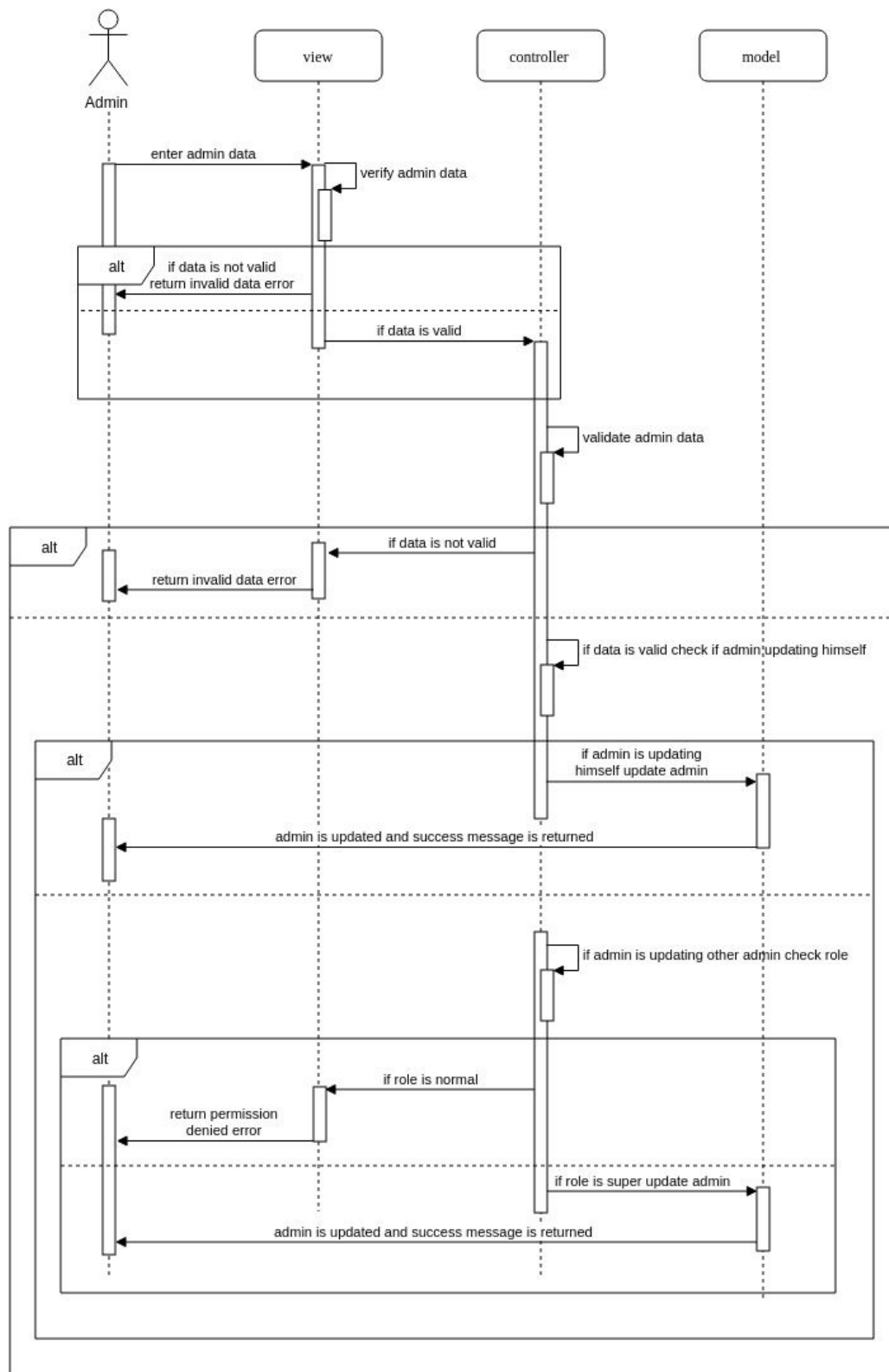


Figure 9: SD7

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

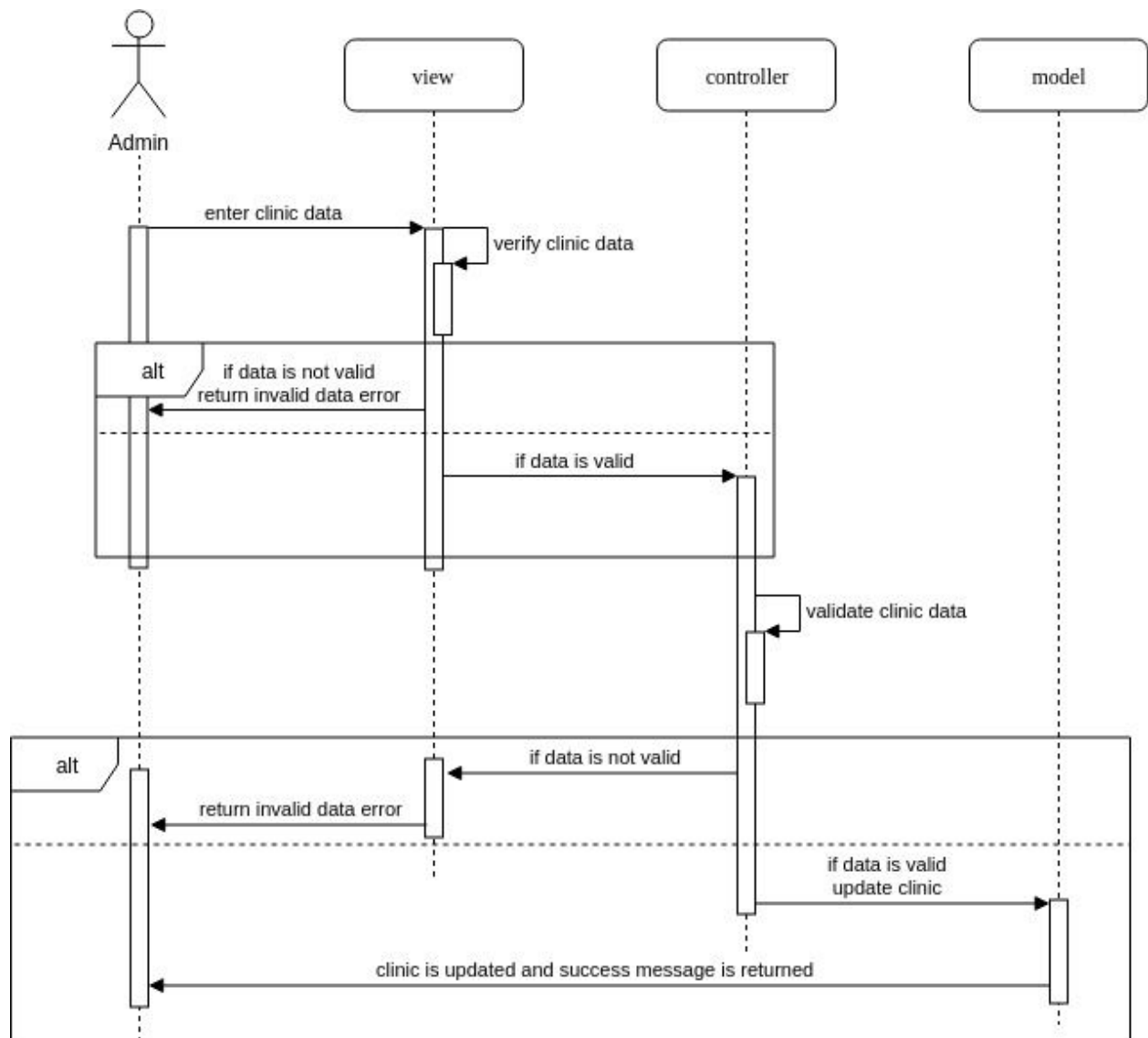


Figure 10: SD8

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

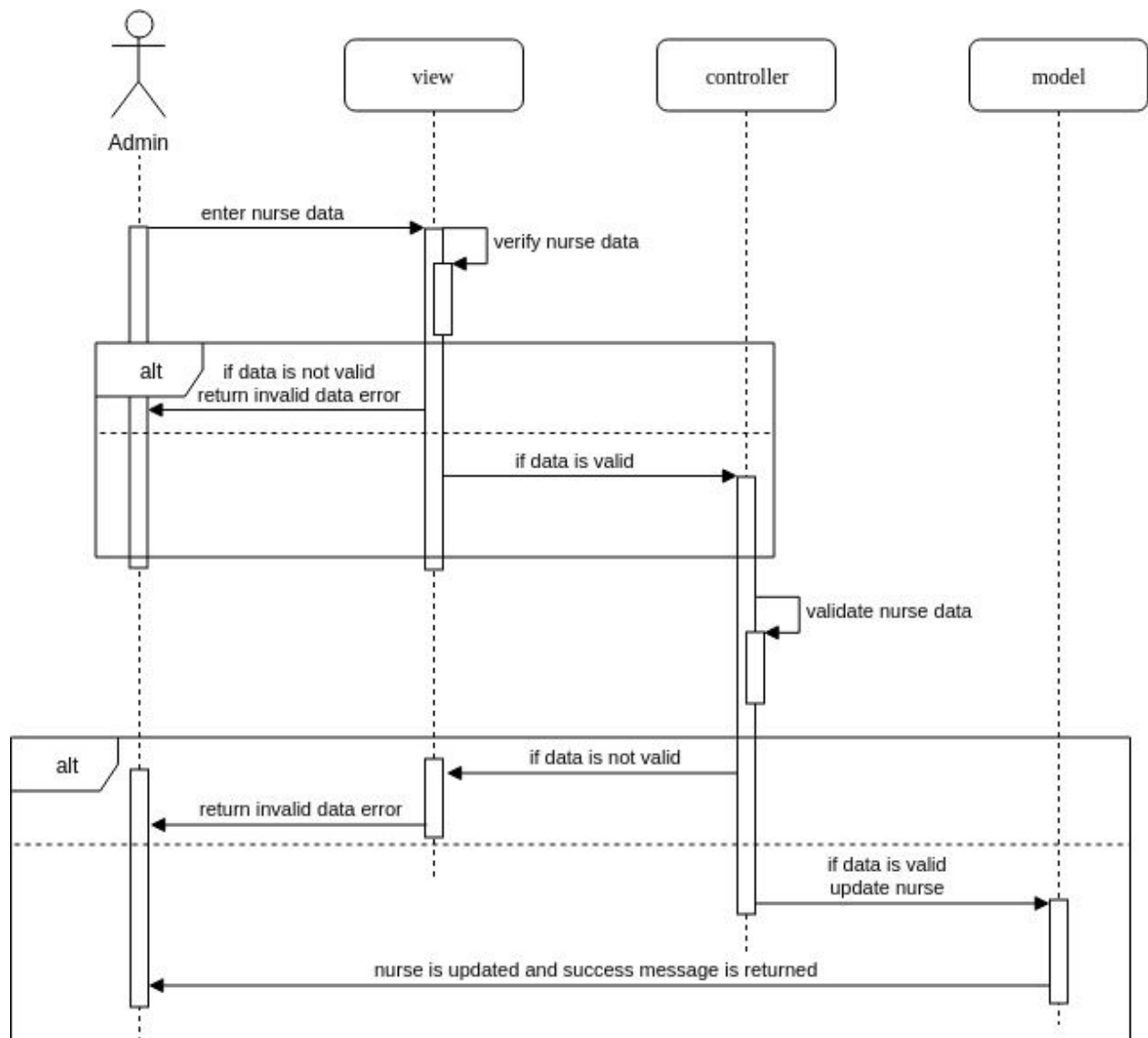


Figure 11: SD9

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

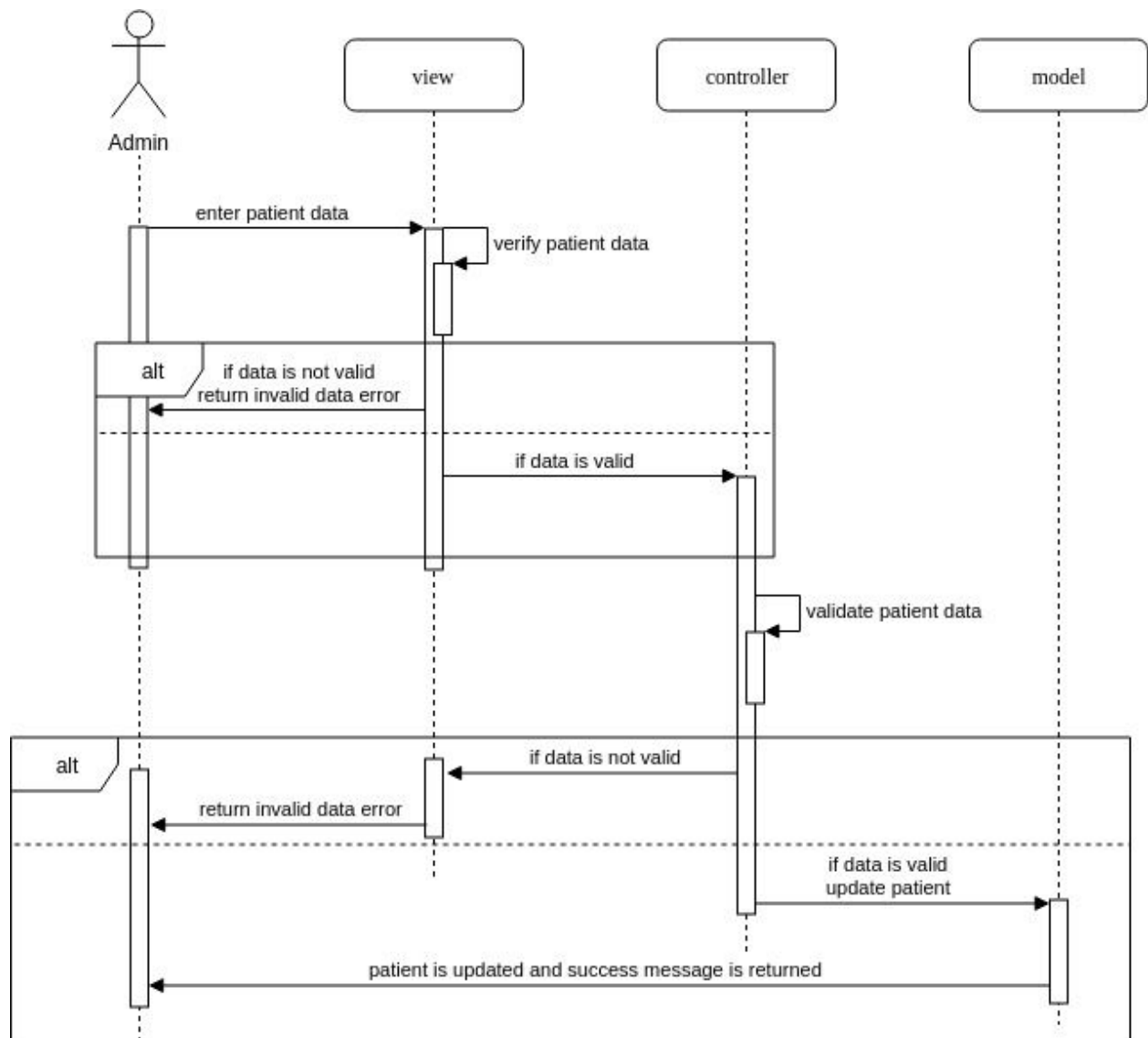


Figure 12: SD10

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

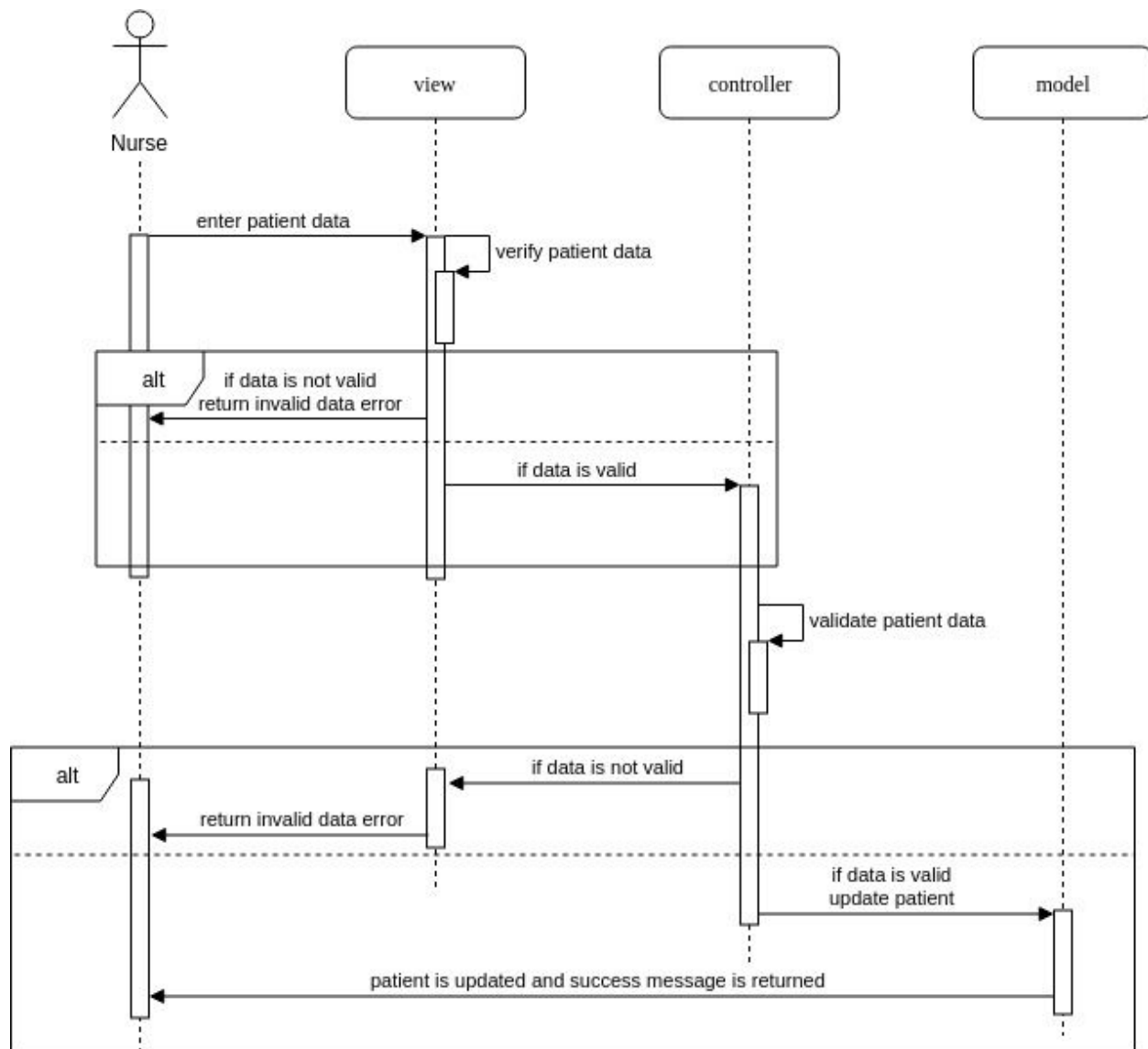


Figure 13: SD11

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

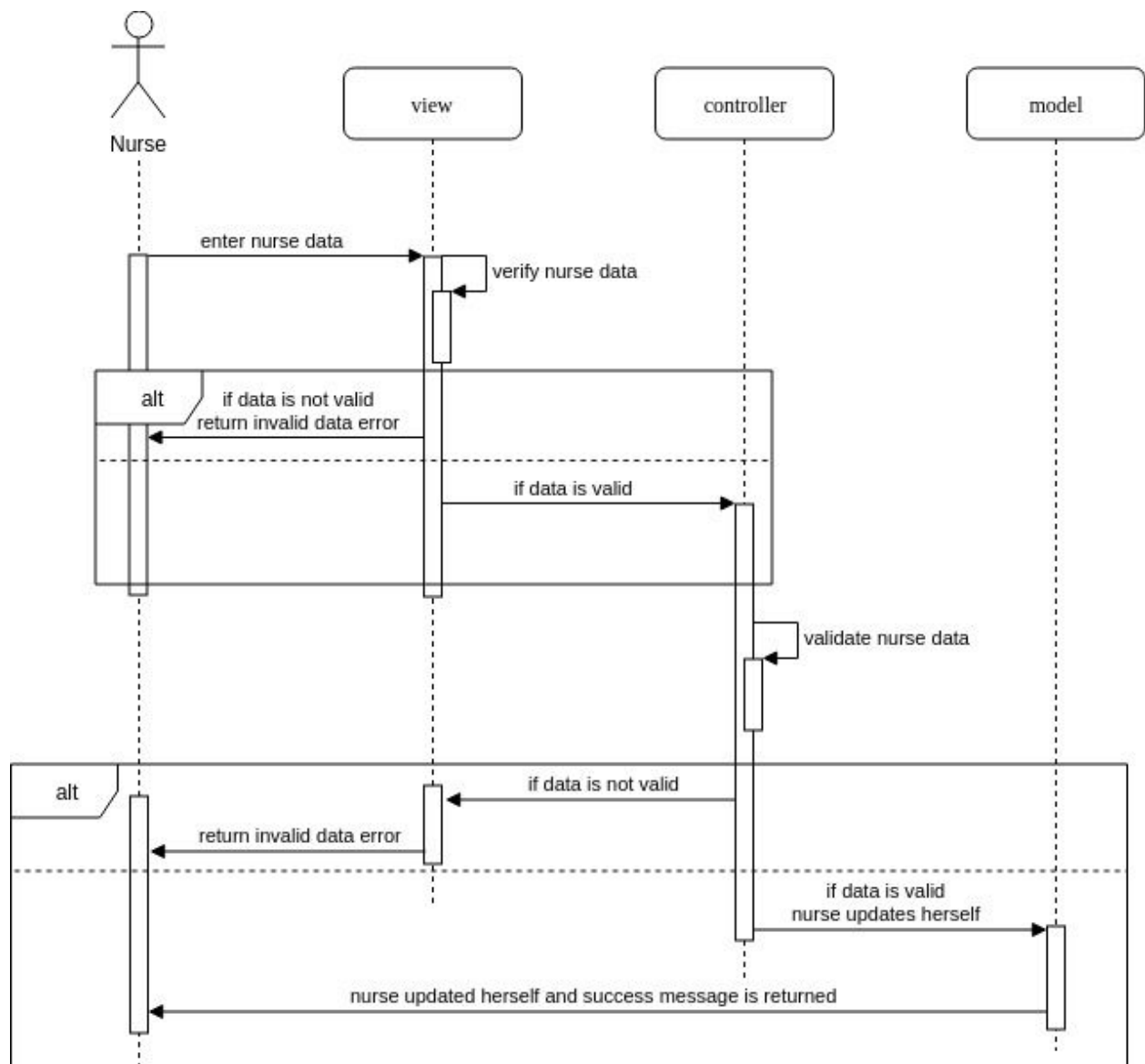


Figure 14: SD12

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

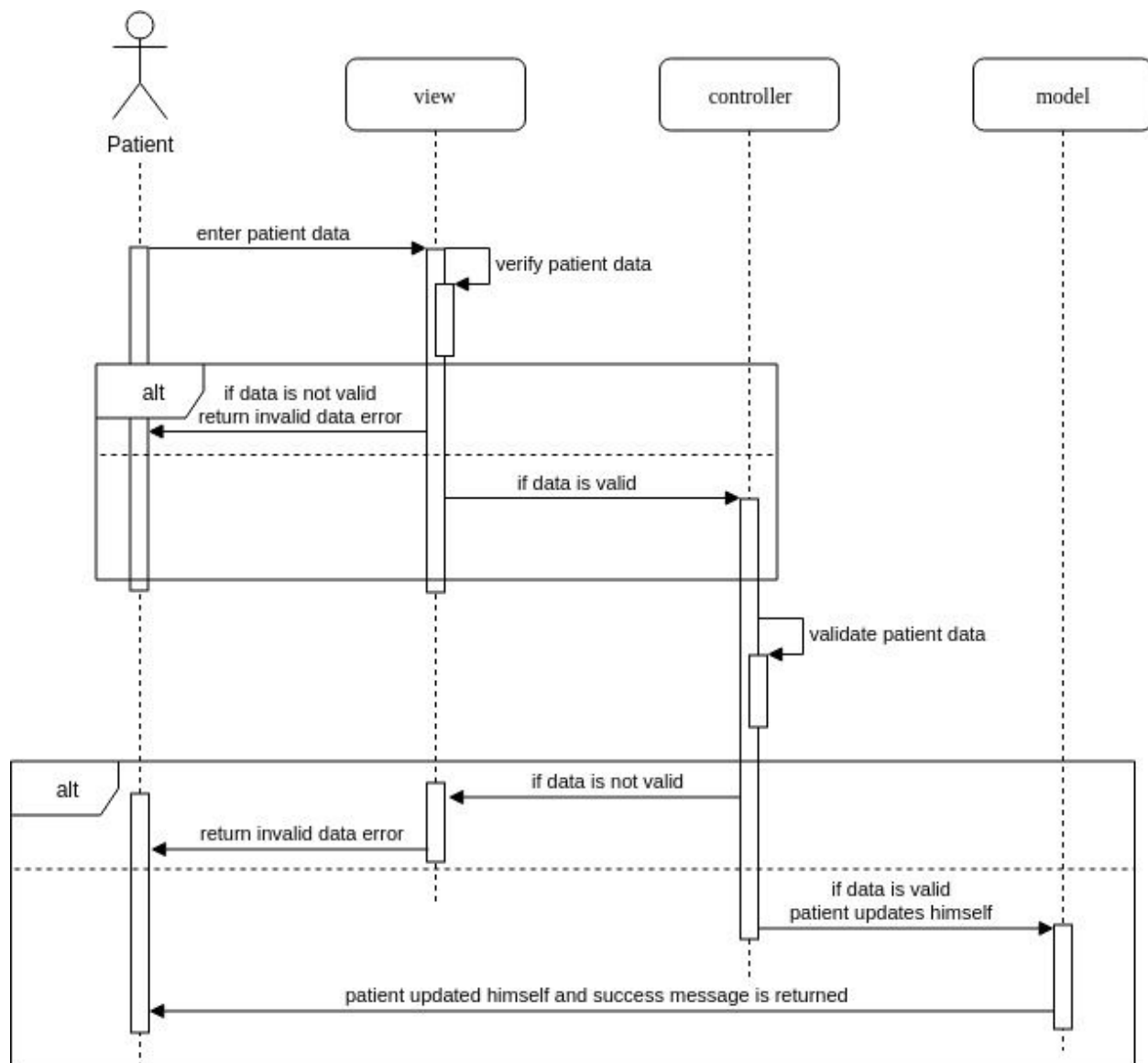


Figure 15: SD13

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

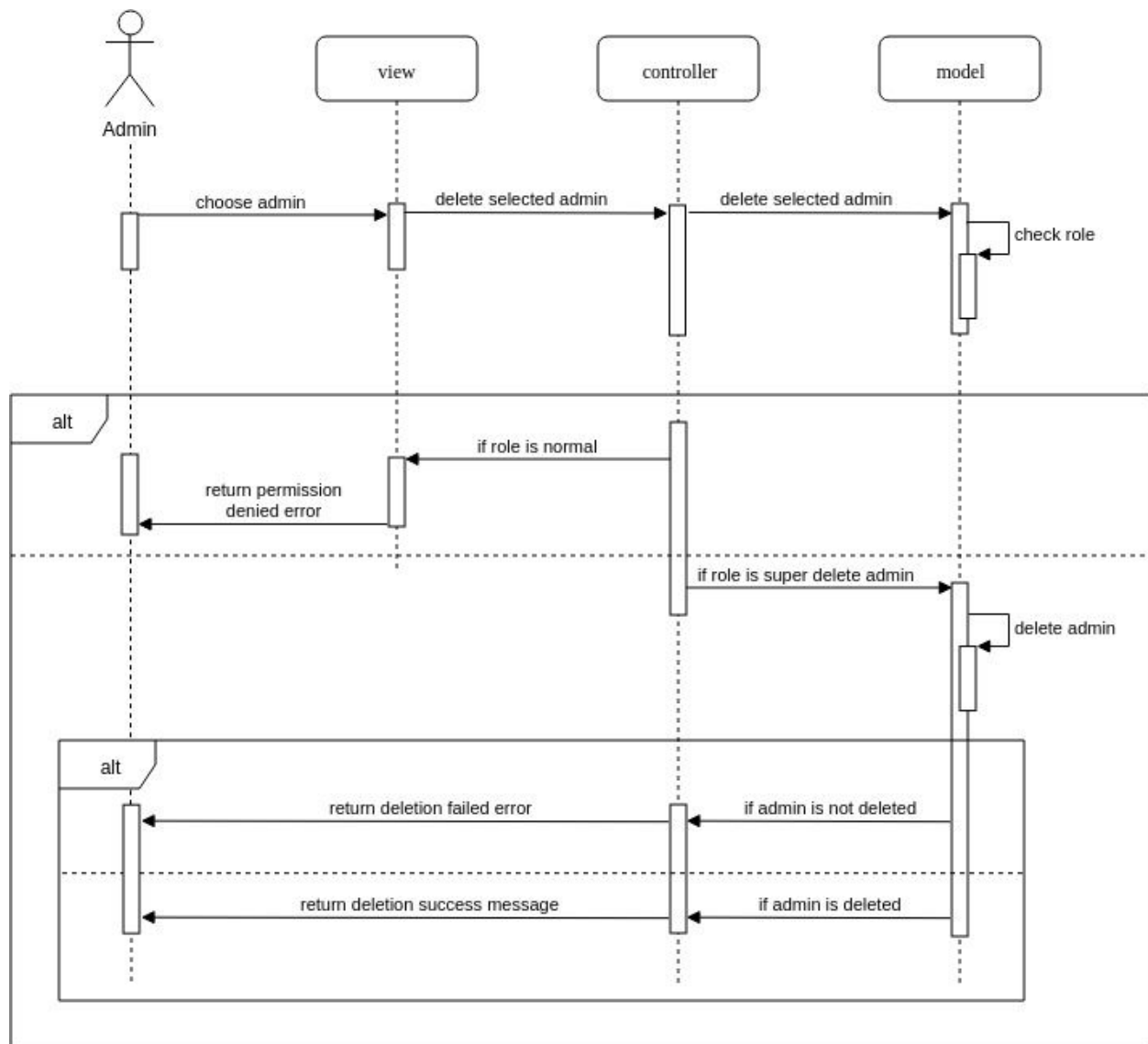


Figure 16: SD14

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

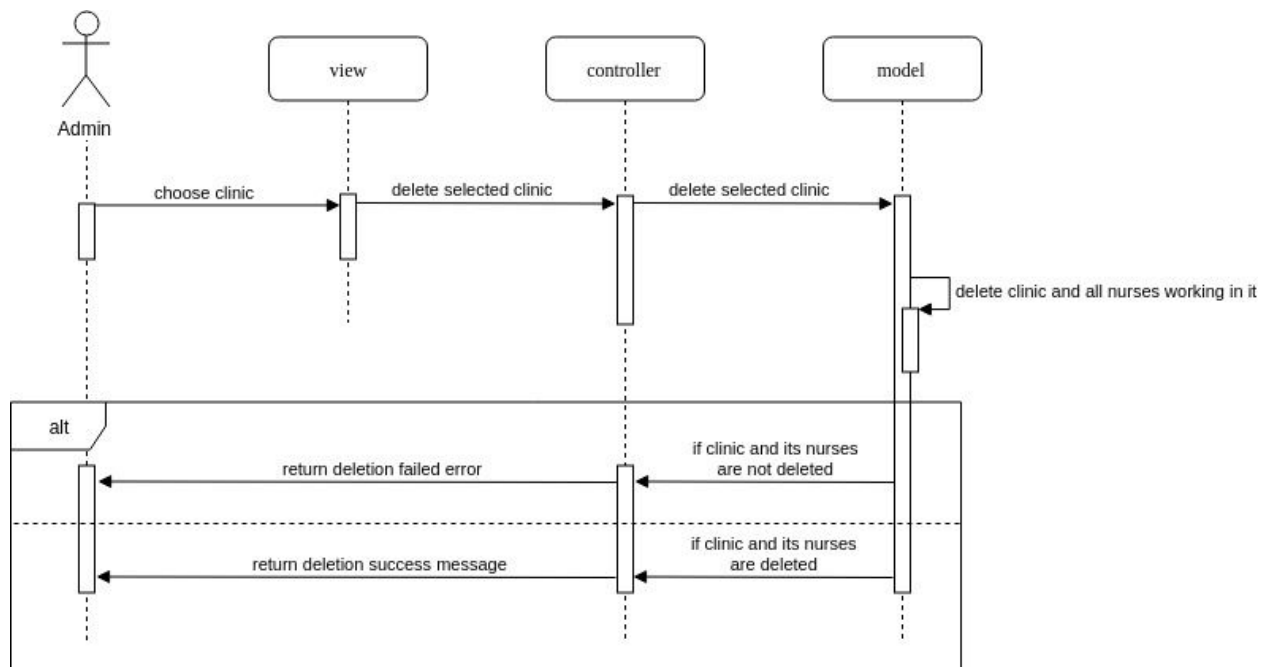


Figure 17: SD15

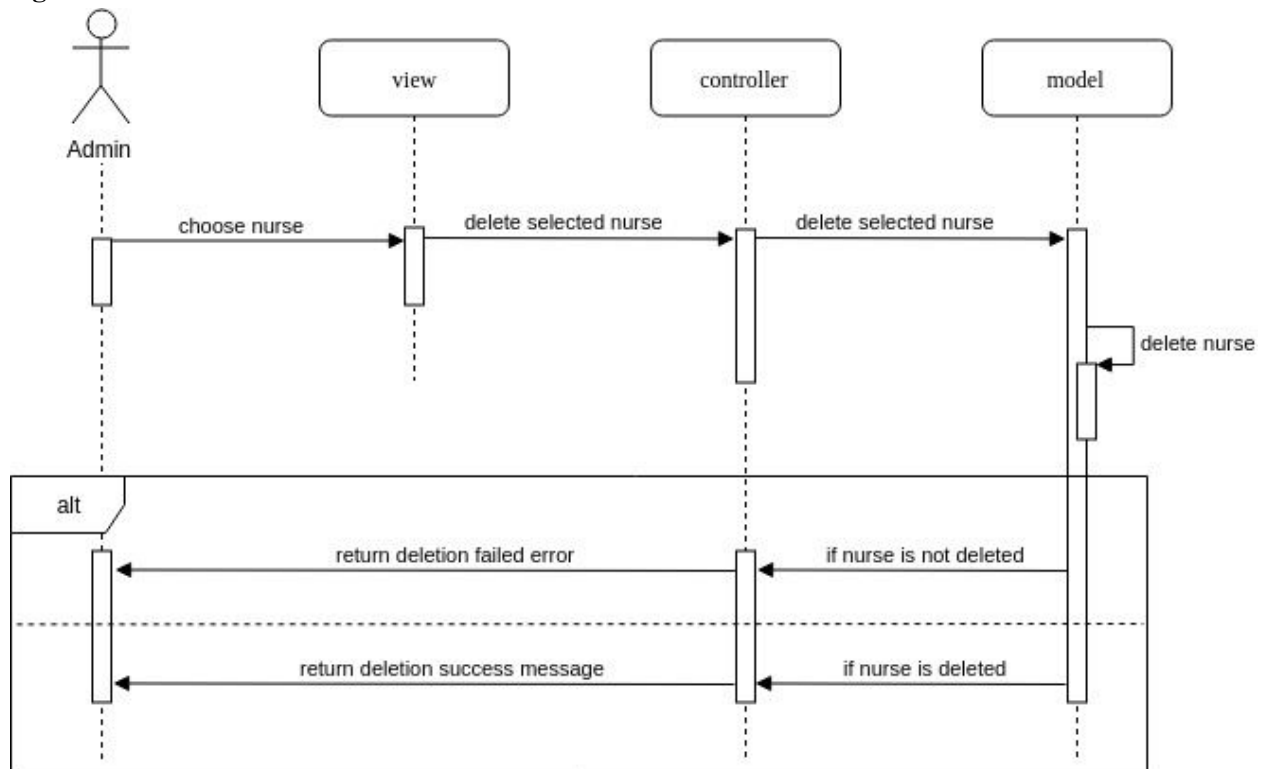


Figure 18: SD16

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

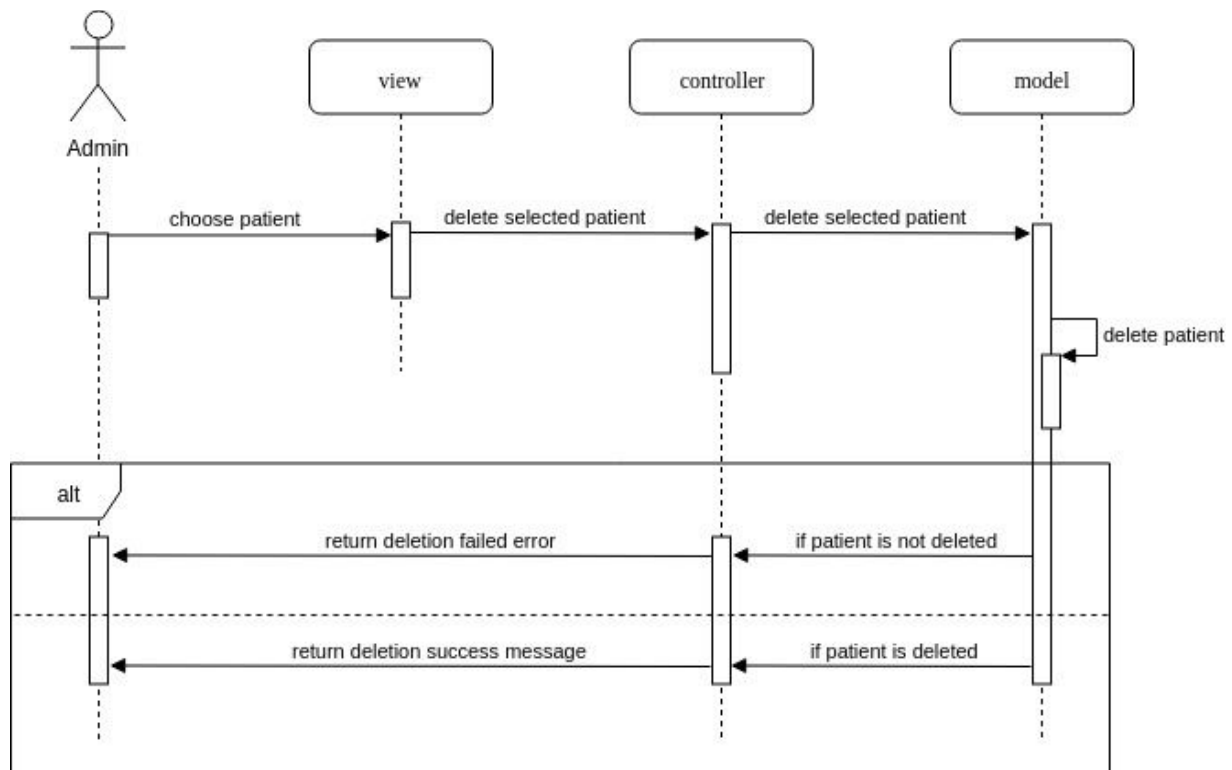


Figure 19: SD17

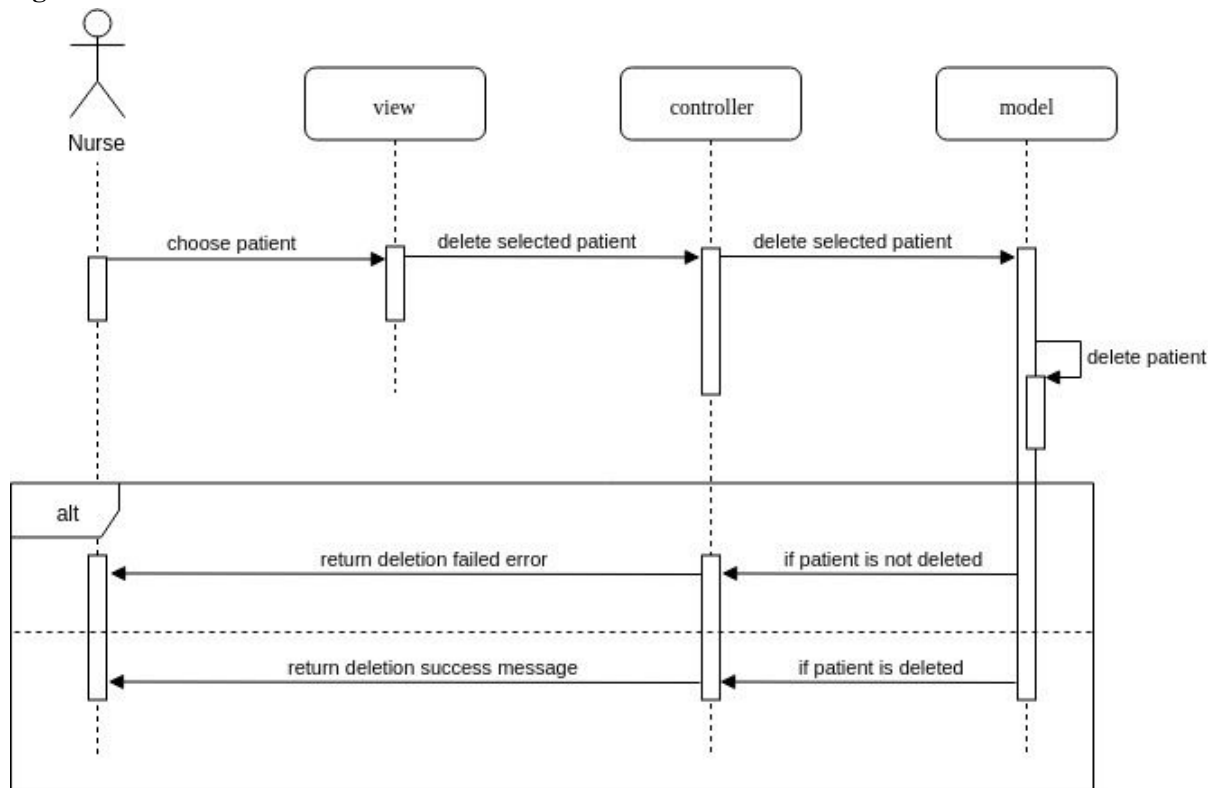


Figure 20: SD18

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

4. Data Models

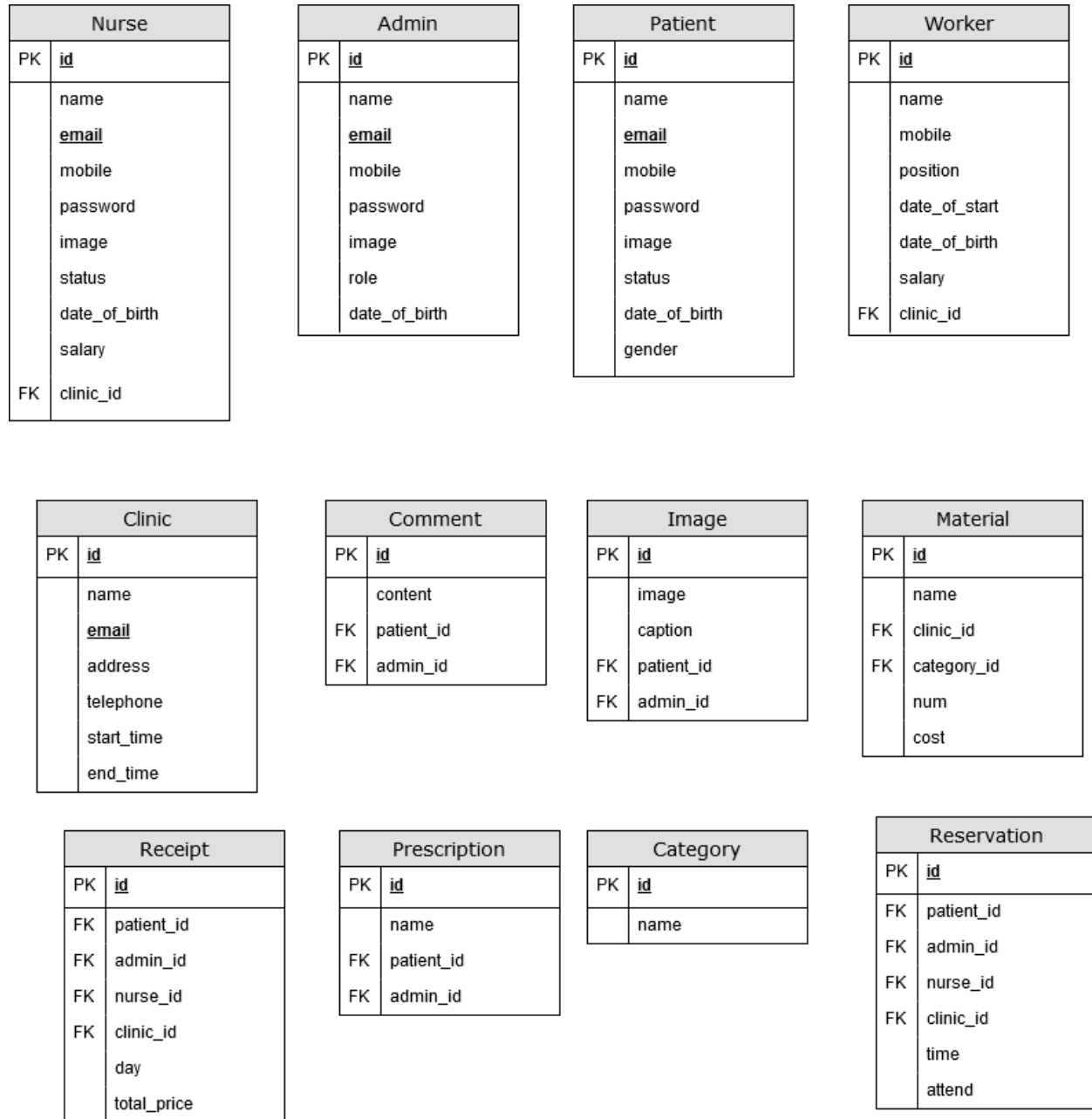


Figure 21 :er-entities

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

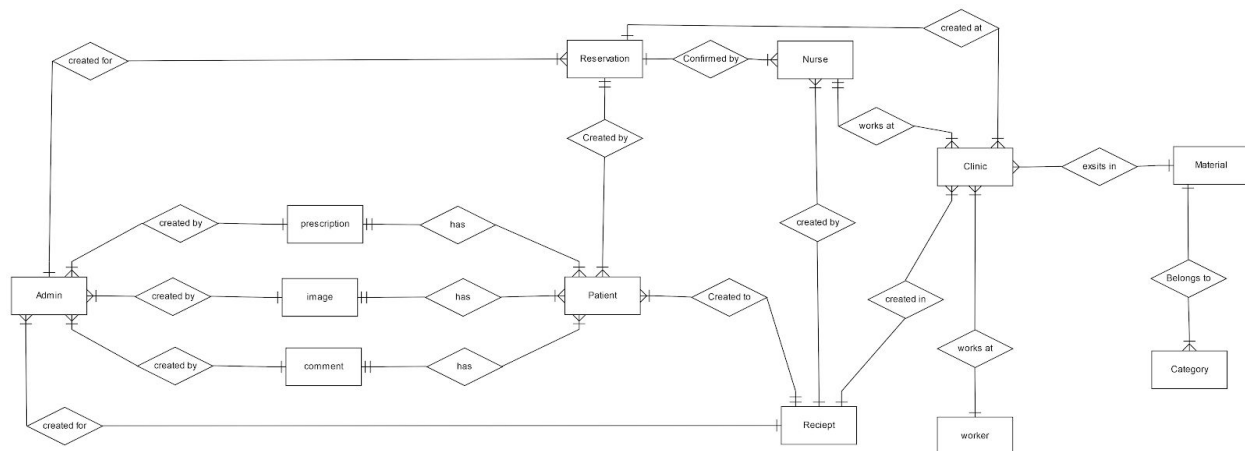


Figure 22 :er-relations

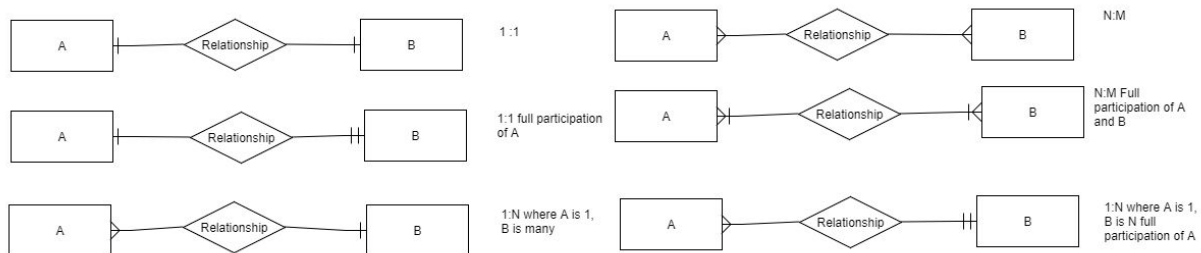


Figure 23 : er-map

Entities

- **Nurse** : assists doctor in running the clinic .
- **Patient** : the main client the system aims to serve .
- **Admin** : a doctor in the clinic (could be an owner or just an assisting doctor) .
- **Clinic** : the place where admins and nurses work .
- **Receipt** : a written statement acknowledging that patients have paid for their treatment.
- **Prescription** : drug description written by doctors for their patients.
- **Comment** : any observations doctors have of their patient's medical condition, they are added to patients' files .
- **Image** : a visual description of a patient's medical condition which is also added to patients' files.
- **Material** : medical substances used by doctors in the treatment and are required to be monitored.
- **Reservation** : a request to have an appointment with a doctor made by patients and approved by nurses.
- **Category** : the classification of materials.
- **Worker** : a person who works at a clinic who isn't a doctor or a nurse but is important to the clinic records.

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

Relationships

- (Reservation-Patient) created by : a list of all reservations made by the patient.
- (Reservation-Nurse) confirmed by : every reservation made by a patient has to be confirmed by a nurse
- (Reservation-Clinic) created at : every reservation is assigned a clinic id to specify the location of the appointment .
- (Reservation- Admin) created for : patients can request a certain doctor while reserving an appointment and so a reservation has information on the doctor it is assigned for .
- (Nurse-Clinic) works at : every nurse is assigned to a certain clinic (only one) .
- (Nurse-Receipt) created by : nurses create receipts for patients after every treatment session to manage the finances of the client and inform their client of their treatment expenses.
- (Receipt-Patient) created to : the receipt holds information about the client it is assigned to.
- (Receipt-Admin) created for : the receipt holds information about the doctor who did the treatment.
- (Receipt-Clinic) created in : every clinic manages its record of receipts independent of other clinics.
- (Clinic-Worker) works at : every clinic employs a number of workers and keeps track of their information.
- (Clinic-Material) exists in : clinics keeps record of their available materials to track their quantity and cost.
- (Material-Category) belongs to : materials that serve a common purpose are classified together under one category to make search and enumeration easy processes.
- (Comment-Patient) has : patients' files contain comments and observations made by doctors on their medical condition.
- (Image-Patient) has : patients' files may contain images that visually aids the doctor to understand the issues of their patients.
- (Prescription-Patient) has : drug prescription given to patients which they can check in their files.
- (Comment-Admin) created by : doctors record their remarks on the medical conditions of their patients in their files.
- (Image-Admin) created by : doctors add images essential to the description of their patient's case to their files.
- (Prescription-Admin) created by : doctors can write drug prescription to their patients for medical purposes.

5. System Deployment

Component diagram (*see section 3.4*) shows the system software integrity as different components of software interact with one another. The following diagram is the deployment diagram which shows the hardware deployment and integrity of the system. Following the two tier client server model the system would be divided into two tiers client tier and server tier.

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

Client tier

This tier is deployed in the users side. It contains views that interact with controllers on server through a browser. This tier sends data and different requests to controllers on the server side.

Server tier

This tier is deployed in the server side. It contains controllers, models and database. This tier receives data and requests from the client side. The connection protocol between the two tiers is a normal interprocess communication (**IPC**).

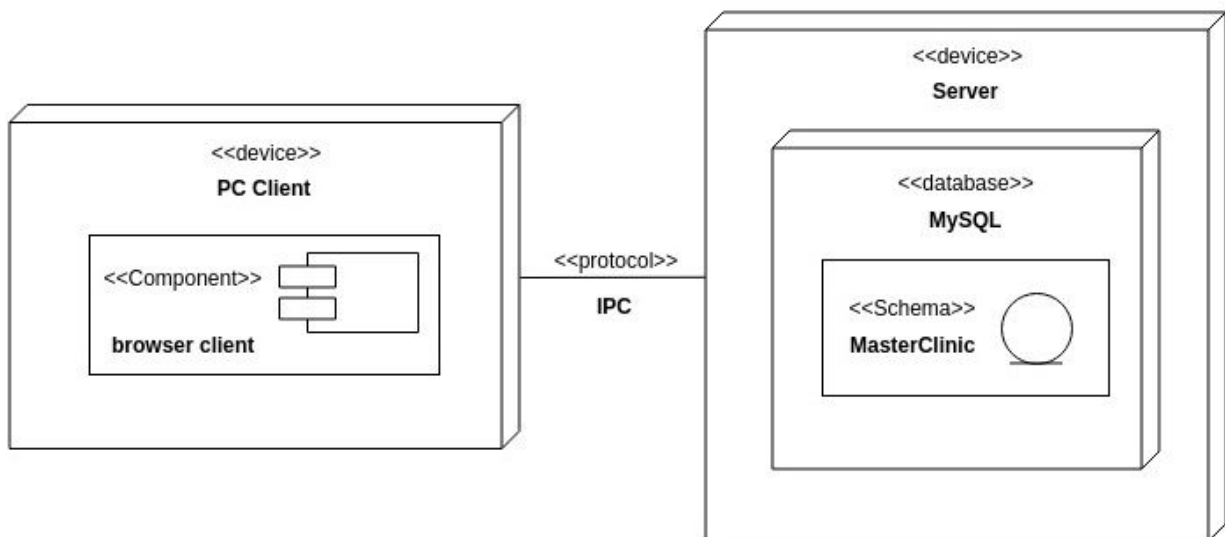


Figure 24: Deployment diagram

6. Traceability to Requirements

	SD 1	SD 2	SD 3	SD 4	SD 5	SD 6	SD 7	SD 8	SD 9	SD 10	SD 11	SD 12	SD 13	SD 14	SD 15	SD 16	SD 17	SD 18
US1	✓	✓	✓	✓					✓			✓				✓		
US2	✓	✓	✓					✓							✓	✓		
US3	✓	✓	✓	✓	✓	✓				✓	✓		✓				✓	✓
US4	✓	✓	✓				✓							✓				
US11	✓	✓	✓	✓	✓	✓												

Master Clinic	CM-identifier: MC_SE02_v1.0
Software Design Specification	Date: 01/04/2018

<This should maintain a matrix similar to the Traceability Matrix in the Software Requirements Specification document. The purpose of this matrix is to map requirements to design elements. This will immediately point out requirements that have not been considered. It will also allow you to trace into the design elements that require to be changed if requirements change. Use requirement IDs as row headers and design element IDs as column headers>