**Appendix 1.5 Scenarios**

**Scenarios**

1. **Add employee scenario:**
2. Manager selects employee type.
3. Manager provides name and relevant information for an employee.
4. System validates employee data by confirming the saving of the entered information.

Classes;

* Manager
* Employee
* Type

Methods:

* Select
* Validate
* Save
* Provide
* Confirming

**B. Search employee scenario:**

1. Manager enters employee name in the system.
2. System returns searched employee.

Classes

-Manager

- Employee

- Database

Methods

* Search

**C. Edit employee scenario:**

1. Manager selects an employee.
2. Manager updates employee information.
3. System saves the updated information.

Classes:

-Manager

- Employee

- System

- Database

Methods;

* Update.

**D. Remove employee scenario:**

1. Manager selects an employee and removes the selected item.System confirms
2. remove has been successful.

Classes:

* Manager
* Employee
* System
* Database

Methods:

* Remove.

**E. Add patient scenario:**

1. Secretary adds patient information and the system validates the information.
2. System saves the information.

Classes:

* Patient.
* Secretary.
* System
* Database

Methods:

* Add
* Validate
* Save

**F. Edit patient scenario:**

1. Secretary selects an patient.
2. Secretary updates patient information.
3. System saves the updated information.

Classes

* Secretary
* Patient
* System

Methods

* Select
* Save
* Update

**G. Remove patient scenario:**

1. Secretary selects a patient and removes patient.
2. System confirms data has been removed.

Classes:

* Secretary
* Patient
* System
* Database

Methods

* Select
* Remove
* Confirm

**H. Search patient scenario:**

1. Secretary enters patient's name in the system.
2. System returns searched patient.

Classes

* Secretary
* Patient
* System
* Database

Methods

* Search.

**I. Remove an appointment scenario:**

1. The secretary finds patient.
2. The system will return a list of patient’s appointments.
3. The secretary picks the desired appointment to remove.
4. The system removes appointment.

Classes

* Secretary
* Patient
* System
* Database
* Appointment

Methods

* Search.
* Remove
* Save

**J. Get a list of appointments scenario:**

1. The secretary enters patient name or/ and doctor name or/ and a period of time.
2. The system will return a list of appointments based on the search criteria.

Classes

* Secretary
* Patient
* Doctor
* System
* Database
* Appointment

Methods

* Search.

**K.  Add a medical case scenario:**

1. Doctor selects patient.
2. The doctor edit’s medical case
3. Doctor updates medical case.

Classes

* Doctor
* Patient
* System
* Database
* Medical case

Methods

* Search.
* Save/Update

**L. Add an appointment**

1. Secretary search patient.
2. Secretary register reason.
3. Secretary search doctors with available dates.
4. Secretary select doctor.
5. System save the information.

Classes

* Secretary
* Patient
* Doctor
* System
* Date
* Database

Methods

* Search
* Register
* Select
* Save

**M. Edit an appointment:**

1. Secretary search patient.
2. Secretary select appointment.
3. Secretary edit appointment
4. System saves the updated information.

Classes

* Secretary
* Patient
* System
* Date
* Database
* Appointment

Methods

* Search
* Edit
* Select
* Save

**N. Renew Medicine:**

1. Doctor search patient.
2. Doctor select medicines.
3. Doctor renew medicine.
4. System saves the updated information.

Classes

* Doctor
* Patient
* Medicine
* System
* Database

Methods

* Search
* Register
* Select
* Renew
* Save

**O. Request Medicine Renew:**

1. Secretary search patient.
2. Secretary select medicines.
3. Secretary send request.
4. System update information.

Classes

* Secretary
* Patient
* Medicine
* System
* Database

Methods

* Search
* Select
* Send
* Update

**P. Approve Medicine Request:**

1. Doctor search patient.
2. Doctor select medicines.
3. Doctor renew medicine.
4. System saves the updated information.

Classes

* Doctor
* Patient
* Medicine
* System
* Database

Methods

* Search
* Register
* Select
* Renew
* Save

**Q. Add Medical Prescription:**

1. Doctor selects the patient.
2. System returns patient list.
3. Doctor chooses the corresponding medicine and adds it.
4. System saves changes.

Classes

* Doctor
* Patient
* System
* Database
* Medicine

Methods

* Select
* Add
* Return

**R. Renew medical prescription:**

1. Doctor searches for patient
2. System returns patients information
3. Doctor selects and edits the existing medical prescription.
4. System saves changes.

Classes

* Doctor
* Patient
* System
* Database
* Medicine

Methods

* Select
* Edit
* Return

**S. Approve renew medical prescription:**

1. System returns  renew medicine requests.
2. Doctor selects one request at a time
3. System returns a list of medicine for the given patient
4. Doctor approves or denies request.
5. System saves changes

Classes

* System
* Doctor
* Patient
* List of medicine

Methods

* Return
* Select
* Approve
* Deny

**T. Add Medicine scenario:**

1. Doctor adds medicine information and the system validates the information.
2. System saves the information.

Classes:

* Medicine.
* Doctor.
* System
* Database

Methods:

* Add
* Validate
* Save

**U. Edit Medicine scenario:**

1. Doctor selects a Medicine.
2. Doctor updates Medicine information.
3. System saves the updated information.

Classes

* Doctor
* Medicine
* System

Methods

* Select
* Save
* Update

**V. Remove Medicine scenario:**

1. Doctor selects a medicine and removes medicine.
2. System confirms data has been removed.

Classes:

* Doctor
* Medicine
* System
* Database

Methods

* Select
* Remove
* Confirm

**W. Search Medicine scenario:**

1. Doctor enters medicine name in the system.
2. System returns searched medicine.

Classes

- Doctor

- medicine

- System

- Database

Methods

* Search.