



[**Introduction 2**](#_gjdgxs)

[**The Small Turtle House Team 2**](#_30j0zll)

[**System Requirements 3**](#_3j2qqm3)

[User Stories 3](#_1fob9te)

[*User Story #1 3*](#_1fob9te)

[**Project Management 3**](#_1y810tw)

[Continuity of Operations Plan (COOP) 3](#_3znysh7)

[Project Plan 4](#_2et92p0)

[*System Architecture Design and Development 4*](#_tyjcwt)

[*System Implementation 0*](#_3dy6vkm)

[Project Postmortem <Postmortem> 0](#_1t3h5sf)

[*Project Wins 0*](#_4d34og8)

[*Root Cause Analysis 0*](#_2s8eyo1)

[*Lessons Learned 0*](#_17dp8vu)

[**System Design 0**](#_3rdcrjn)

[System Architecture <Milestone #1> 0](#_26in1rg)

[*Component Design 0*](#_lnxbz9)

[*Data Flow 0*](#_35nkun2)

[System Components <Milestone 2> 0](#_1ksv4uv)

[*Component [Component Name 1] 0*](#_44sinio)

[*Component [Component Name 2] 0*](#_2jxsxqh)

[*Component [Component Name n] 0*](#_z337ya)

[Project portfolio template directives and placeholders (delineated by “[ ]” or “< >” and/or highlighted or optional sections not included) should be removed from the document prior to submission. Empty sections for inclusion in later submissions may remain in the document for early submissions.]

[IMPORTANT: All diagrams developed using Enterprise Architectures must include the following acknowledgement: “Thanks to SPARX Systems for LSU student and faculty use of Enterprise Architect for academic purposes”.]

# Introduction

TurtleMD can be used by hospitals or doctor’s offices to allow patients to schedule appointments and communicate with their physicians. Before coming to each appointment, patients fill out a questionnaire that provides their physician with symptoms so that doctors can more efficiently make a diagnosis.

Doctors would be able to see their upcoming appointments, reason for each visit, patient details and history, and a list of possible diagnoses based on the information provided by the patient.

TurtleMD will allow patients to make changes to appointments as well as view test results with an evaluation.



Core Features:

* Symptom forms for patients to fill out
* Highlighted possible conditions recommended to doctors based on forms
* Interactive appointment scheduling
* Medical records

Viable Features:

* Payment management
* Medication recommendations
* Predictive model for diagnoses

# The Small Turtle House Team

Github: <https://github.com/zanwar2/Small-Turtle-House>

* Ziyad Anwar – **Leader**, component diagram management,
* Tyler Vu – component diagram
* Matthew Kleimeyer – component diagram, portfolio management
* James Fazio – data flow diagram management,
* Jesse Chang – data flow diagram
* Jacob Edwards – data flow diagram, code implementation skeleton author

## User Stories

*As a patient, I want to submit my symptoms to my primary care before going in for a visit, so my doctor will have knowledge of my issues before I get there.*

*As a doctor, I want to know my patient’s symptoms before the visit, so I can provide a more accurate diagnosis and allow for treatment plans to be set up more easily*

## Continuity of Operations Plan (COOP)

-If someone is unable to make a meeting, the Team Leader (currently Ziyad Anwar) will catch them either at a separate time or through an Email/discord call where they privately discuss everything that needs to be discussed. If anything needs to be brought up to the group a necessary meeting will be scheduled, and everything will be discussed there.

-If someone were to drop from the group temporarily, the work that the member will not be able to do will be determined and be evenly distributed to others. Once they return, they will be caught up as much as possible and be put right back to work as soon as they are able to get back to work.

-If someone were to drop from the group permanently, the work will be distributed evenly unless someone is willing to pick up more slack then others. If necessary, the current team leader will pick up more slack then necessary.

Project Plan

System Architecture Design and Development

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
| **1** | Conceive general components of architecture |  | 1 | 1 | 2/8 | 2/10 | 2/10 | 2/11 |
|  |  |  |  |  |  |  |  |  |
| **2** | Design component diagram | 1 | 5 |  | 2/10 | 2/11 | 2/10 | 2/11 |
| 2.1 | Subcomponents | 2 | 3 |  | 2/10 | 2/11 | 2/11 | 2/11 |
| 2.2 | Finalize diagram | 2.1 | 2 |  | 2/10 | 2/11 |  |  |
|  |  |  |  |  |  |  |  |  |
| **3** | Convert component diagram to data flow diagram | 2.2 | 2 |  | 2/10 | 2/11 | 2/11 | 2/11 |
| 3.1 | Finalize diagram | 3 | 2 |  | 2/10 | 2/11 | 2/11 | 2/11 |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

### System Implementation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
| **1** | Conceive Classes/Files for each component |  | 1 |  | 2/11 | 2/15 | 3/14 | 3/17 |
| 1.1 | Main components |  | 1 |  | 2/11 | 2/15 | 3/14 | 3/14 |
| 1.2 | Sub components | 1.1 | 1 |  | 2/11 | 2/15 | 3/14 | 3/16 |
| 1.3 | Databases |  | 1 |  | 2/11 | 2/15 | 3/14 | 3/17 |
| **2** | Write methods and private variables for each component |  | 3 |  | 2/11 | 2/15 | 3/22 | 3/28 |
| 2.1 | Private variables |  | 3 |  | 2/11 | 2/15 | 3/22 | 3/28 |
| 2.2 | Methods |  | 3 |  | 2/11 | 2/15 | 3/22 | 3/28 |

## Project Postmortem <Postmortem>

### Project Wins

[Provide a bulleted list of at least 3 positive aspects of the project.]

### Root Cause Analysis

[Provide a bulleted list of at least 3 negative aspects of the project. For each negative, provide the answer to the three successive “Why” questions. ]

### Lessons Learned

[For each negative aspect identified in the Root Cause Analysis, provide a mitigation strategy (i.e., what process should be introduced) to ensure that the problem is not repeated in subsequent projects.]

# System Design

Our design utilizes JavaFX FXML files controlled by Java “controller” classes to interface with GUI features. These GUI actions cause the code to communicate with the SQL databases.

## System Architecture

Our system architecture focuses on the two different types of users: patients and staff. These two users interface with the same data in different ways.

### 

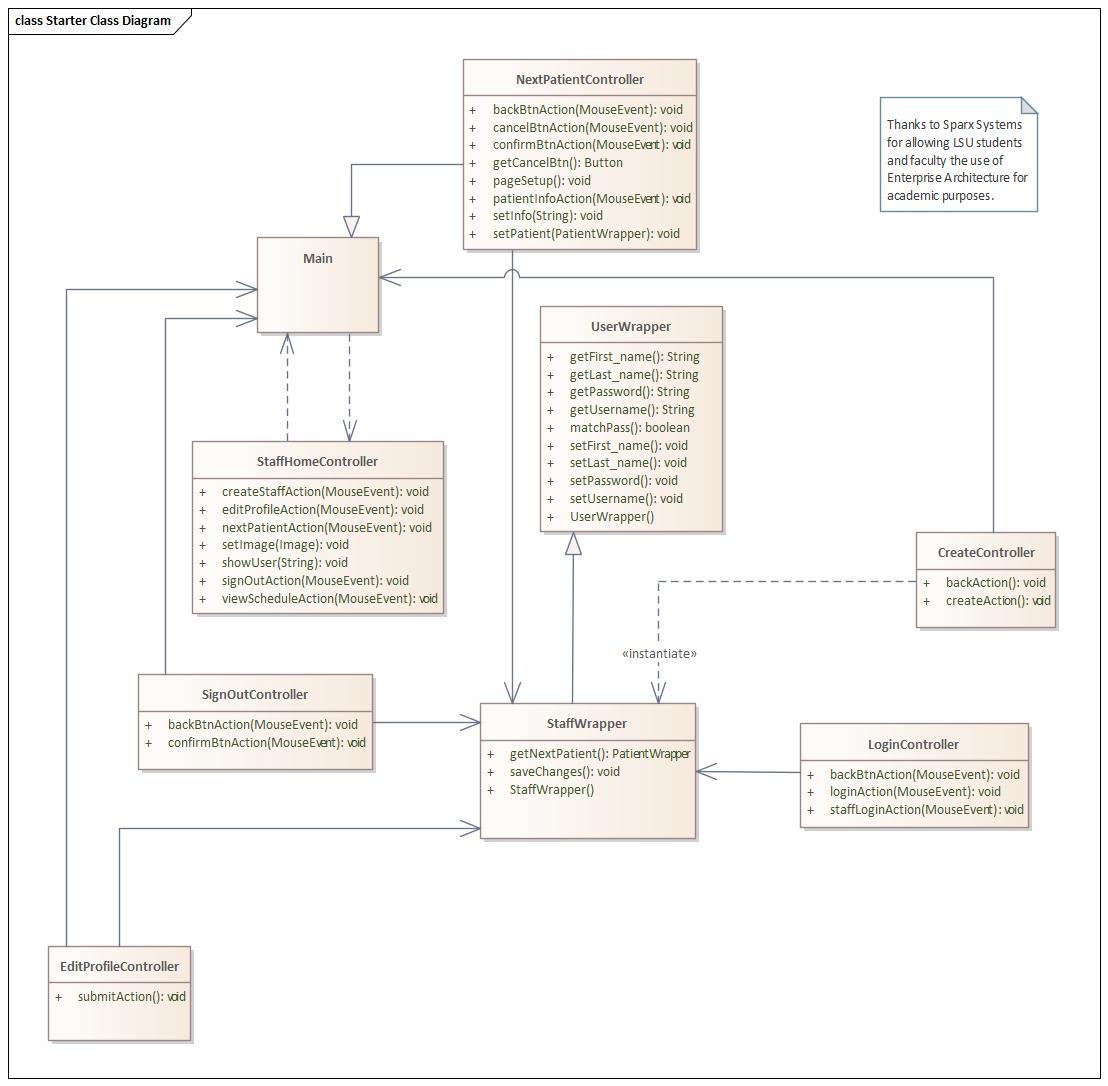
### Component Design

### 

Data Flow

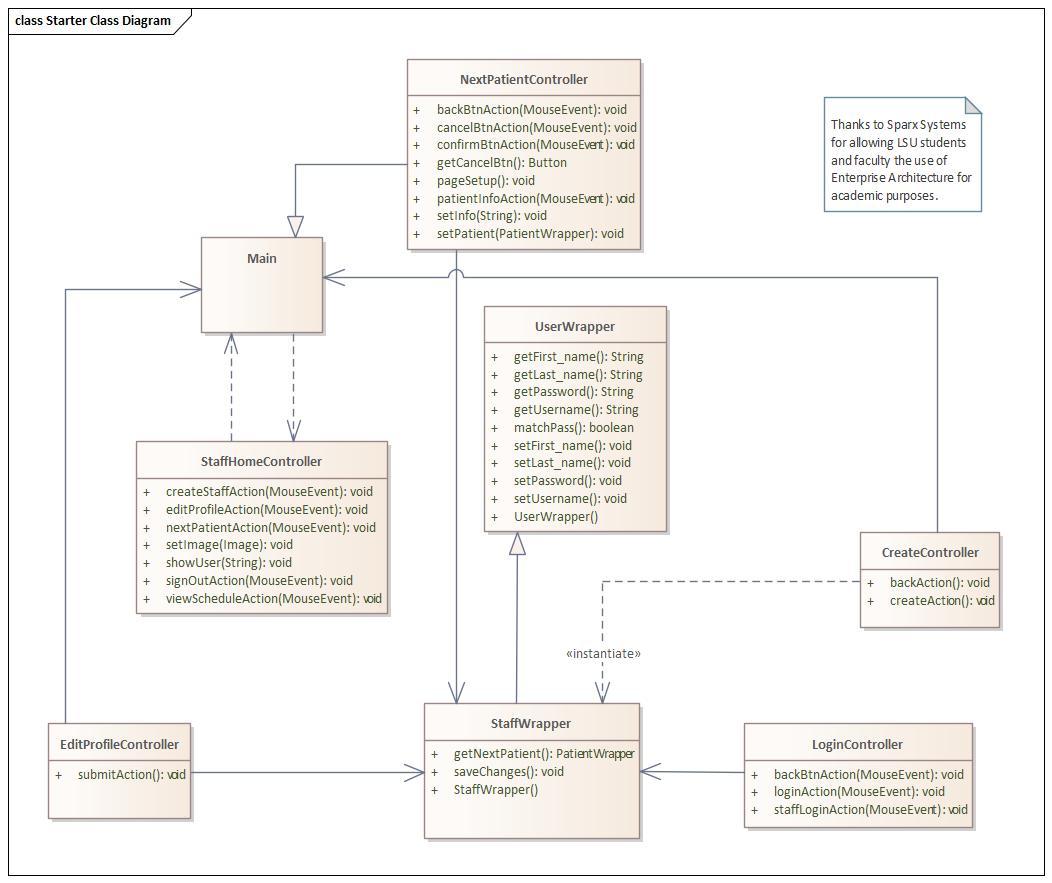
## System Components

### Component **Patient**

The patient user portal

Component **Medical Professional**

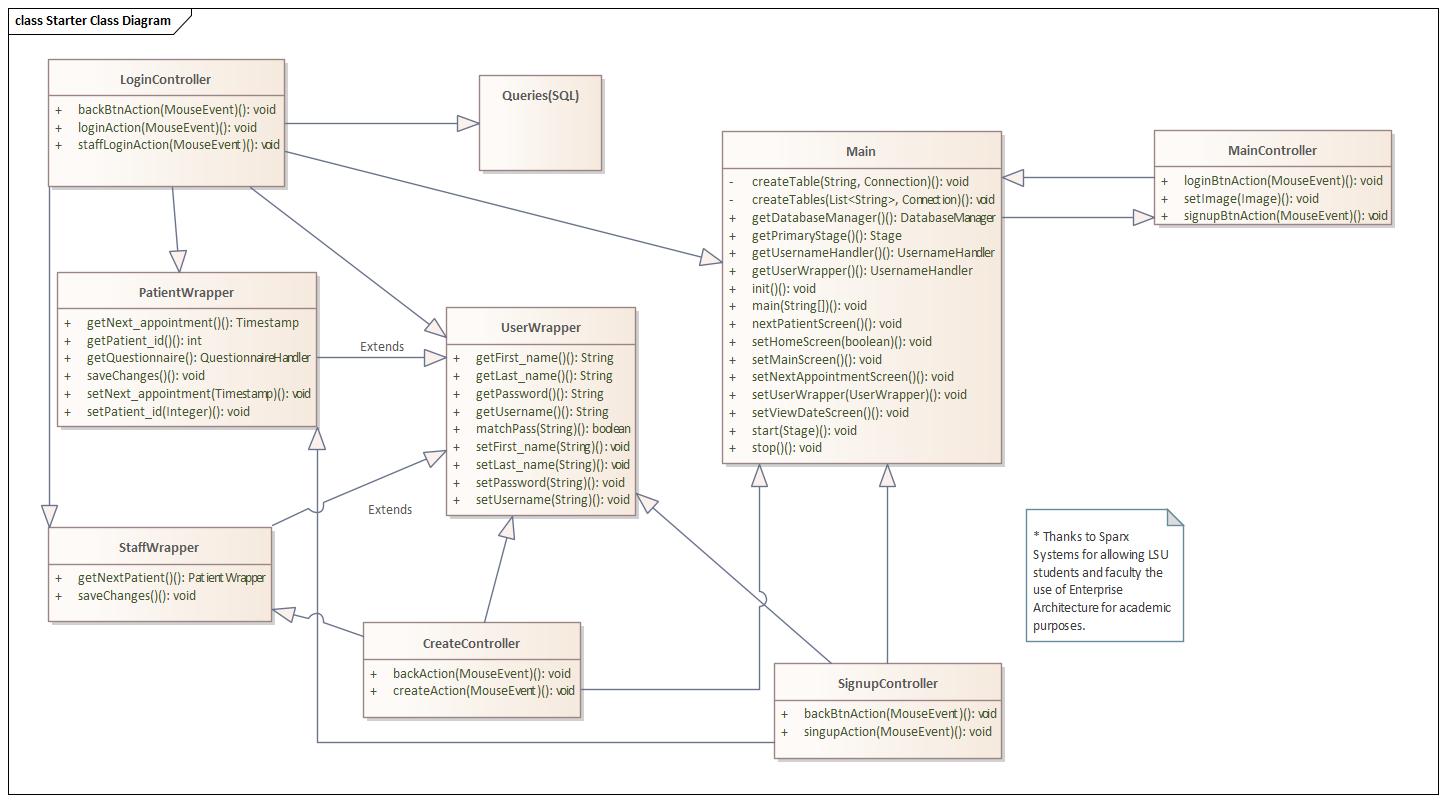
The staff user portal



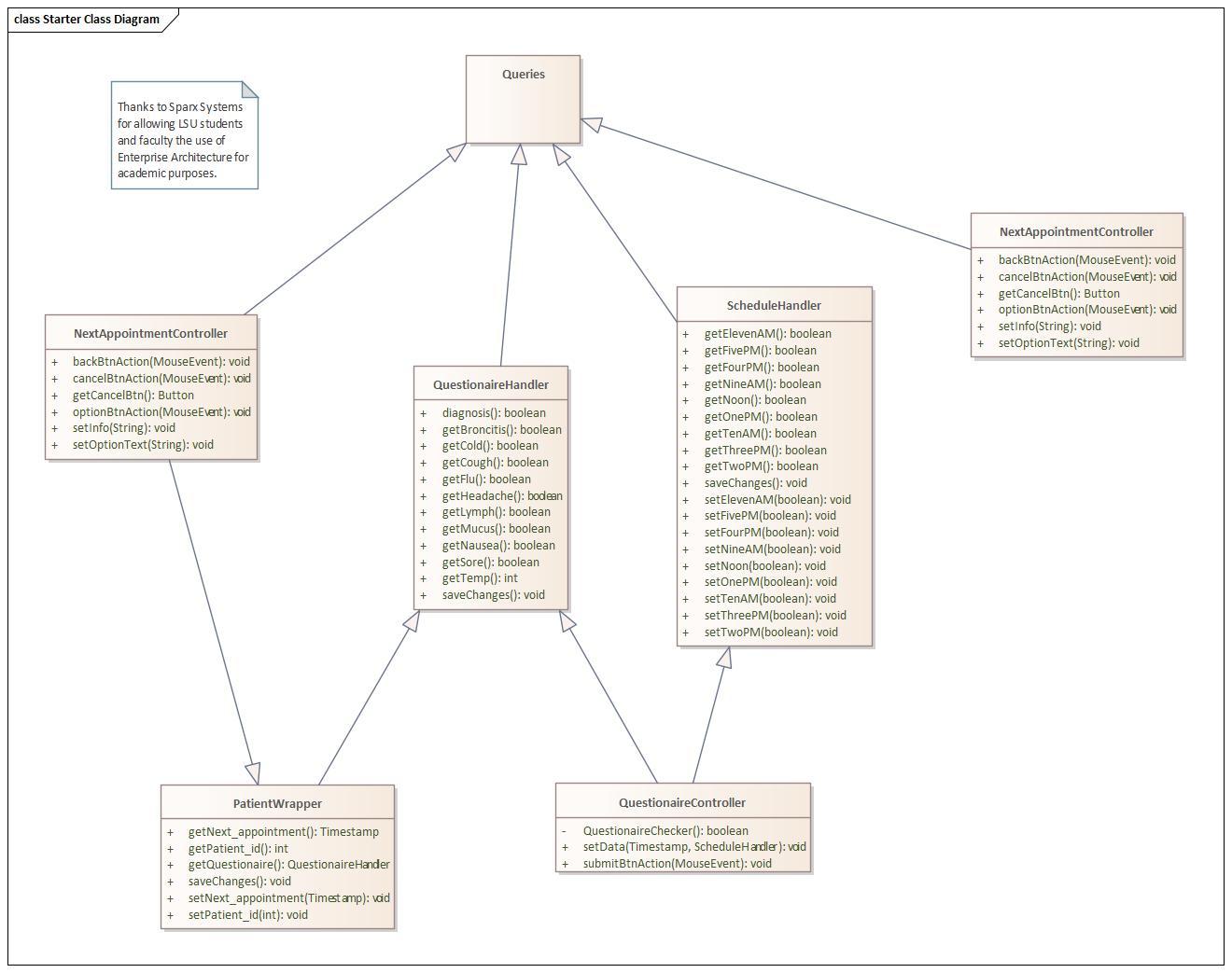
### 

### Component User Interface

The graphical user interface



Component Appointment Detailer

Deals with patient’s data and questionnaires for symptoms

Component Schedule Integration

Deals with scheduling between doctors and patients

