SDS Document

Winter 2018

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue** | **Description** | **Author** |
| 11/28/17 | 1.0 | Initial Revision | Everyone |
| 11/30/17 | 1.1 | Revision of Entity Classes | Everyone |
| 11/30/17 | 1.2 | Revision with Class Diagram | Everyone |
| 12/04/17 | 1.3 | Final Revision of Entire Document | Everyone |
| 01/29/18 | 1.4 | Change to SDS, providing required sections | Jak, Andy, Sam |
| 01/29/18 | 1.5 | Edited written sections, added GIT Branch Table | Stephanie |
| 03/05/18 | 1.6 | Added the Updated Contribution section | Everyone |
| 04/04/18 | 1.7 | Added all sections for Deliverable 4 | Sam |

Table of Contents

[Potential Entity Classes 4](#_Toc505024865)

[Classes Attributes 5](#_Toc505024866)

[Entity Class Diagram (revised) 9](#_Toc505024867)

[Use Case Diagram (revised) 10](#_Toc505024868)

[Revision Explanation 11](#_Toc505024869)

[User GIT Branches 12](#_Toc505024870)

# Potential Entity Classes

|  |  |
| --- | --- |
| Class name | Brief Description |
| User | The base user class of the Self Start system, of which the three specialized users inherit from. |
| Patient | An entity class which holds all the relevant information for the client account within Self Start. |
| Physiotherapist | An entity class which holds all the relevant information for the physiotherapist account within Self Start. |
| Administrator | An entity class which holds all the relevant information for the administrator account within Self Start. |
| Payment | A class which holds the patient’s payment information and history within it. |
| Appointment | An entity for the appointments scheduled that the patient attends. |
| Treatment | An entity class which contains the patient’s current treatment progress as well as their past treatment details. |
| Exercise | An entity class which stores the information for a specific exercise and the associated information. |
| Assessment | An entity class for the differing tests which the client can perform to denote their treatment progression. |
| Rubric | The rubric entity class is a container for several rubric questions that the user can fill out during their treatment. |
| RubricQuestion | Specific answerable questions pertaining to the rubrics of which they are assigned. |
| Plan | The general or custom rehabilitation plans that are contained within the Self Start system |
| DynamicForm | An entity class for a dynamic form on the site, which contains specific questions. This entity class is maintained by the administrator. |
| DynamicFormQuestion | Specific answerable questions pertaining to the forms of which they are assigned. |

# Classes Attributes

1. ***User* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Username | String | General username field |
| Password | String | General password field |

1. ***Patient* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Username | String | Patients username |
| Password | String | Patient’s password stored using a safe hashing algorithm |
| Name | String | Identifies the user based off their name |
| Gender | Enum | Stores the preferred gender of the user (male, female, prefer not to say) |
| DOB | Date | The patients date of birth |
| Address | String | The patients current place of residence |
| Telephone Number | String | The preferred contact number for the patient |
| Health Card Number | String | The patient’s health card information |
| Marital Status | Enum | Stores the marital status of the patient (married, single, divorced, widowed, prefer not to say) |
| Occupation | String | What the patient does for a living |

1. ***Physiotherapist* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Username | String | The identification number specific to the physiotherapist |
| Password | String | Physiotherapist’s password stored using a safe hashing algorithm |
| Name | String | The physiotherapists name |
| Email | String | Physiotherapists email |
| Phone Number | String | Physiotherapists phone number |

1. ***Administrator* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Username | String | Administrator username |
| Password | String | Administrators’ password stored using a safe hashing algorithm |

1. ***Payment* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Credit Card Number | String | Patients credit card number |
| Expiry Date | String | Patients credit card expiry date |
| CVV | Int | Patients CVV number on the back of card |
| Payment Date | Date | The day the patient paid using their credit card |
| Amount | Float | The amount the patient paid |

1. ***Appointment* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Appointment Date | Date | The date of the appointment and time |
| Duration | String | The duration of the appointment |

1. ***Treatment* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Name | String | Name of the specific treatment |
| Date | Date | Date patient received treatment |
| Description | String | The specific description of the treatment |
| Classification | String | Classifying the patient’s specific ailments (i.e. shoulder, knee, etc.) |

1. ***Exercise* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Unique Identification Code | Int | The identifier for the specific exercise |
| Name | String | The name of the treatment plan |
| Description | String | A brief description of what the plan entails |
| Author Name | String | The name of the physiotherapist that created the plan |
| Objectives | String | Goals for the patient |
| Action Steps | String | Certain milestones the patients treatment |
| Location | String | Exercise’s specific location |
| Frequency & Duration | String | Specific reps and sets for the exercise |
| Target Date | Date | Specific date for completion of exercise |
| Multimedia URL | String | Specific URL for the video of the exercise being completed |

1. ***Assessment* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Unique Identification Code | Int | The identifier for the specific assessment |
| Name | String | The name of the assessment test |
| Description | String | Short description of the assessment test |
| Author Name | String | Name of the author |
| Assessment Tool | String | Description of assessment tools used |

1. ***Rubric* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Unique Identification Code | Int | The identifier for the specific rubric |
| Format | String | A description of the format the assessment the rubric is associated to |
| Rating | Int | Between one and ten based off how the user feels |

1. ***RubricQuestion* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Question | String | The specific question to the rubric it exists within. |
| Answer | String | The users answer to the rubric question |

1. ***Plan* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Unique Identification Code | Int | The identifier for the specific treatment plan |
| Name | String | The name of the treatment plan |
| Description | String | A brief description of what the plan entails |
| Author Name | String | The name of the physiotherapist that created the plan |
| Overall Rehabilitation Goal | String | A short description of the overall treatment goals |
| Timeframe | String | Certain progress dates for the patients treatment |

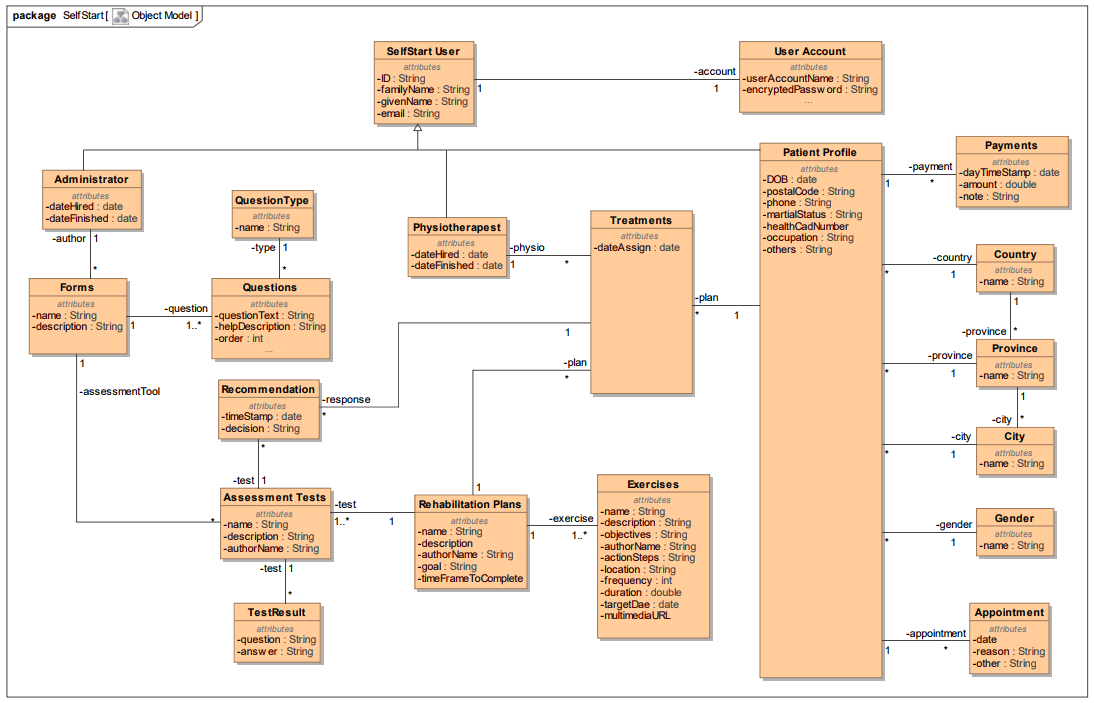
1. ***DynamicForm* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Name | String | Specific name of the dynamic form for identification reasons |
| Classification | String | An identifier pertaining to the part of the Self Start system that the dynamic form belongs to |
| Description | String | Displays some simple information about the form and its field |

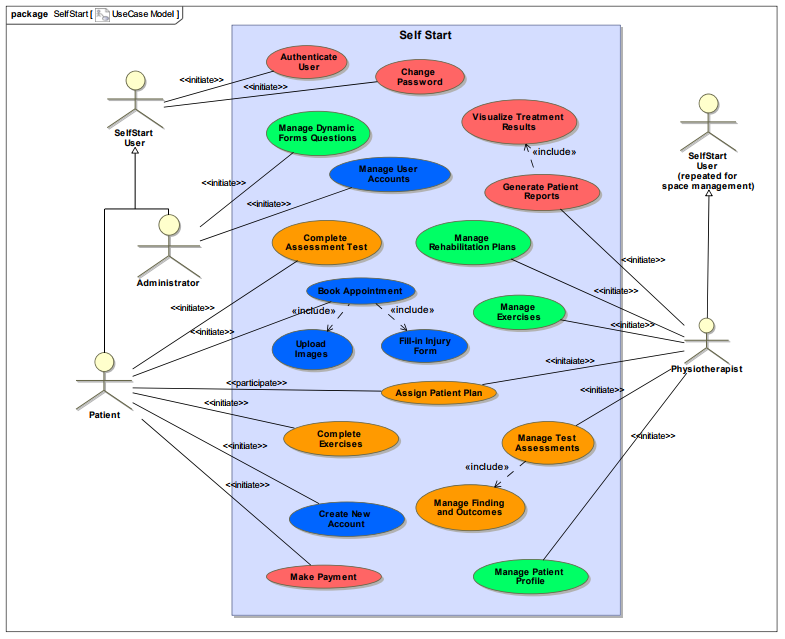
1. ***DynamicFormQuestion* class**

|  |  |  |
| --- | --- | --- |
| Attribute name | Type | Brief Description |
| Question | String | The specific question to the dynamic form it exists within. |
| Answer | String | The users answer to the dynamic form question |

# Entity Class Diagram (revised)



# Use Case Diagram (revised)



# Revision Explanation

For our updated version of the SDS document, we decided upon replacing our original Use Case and Entity Class diagrams with the provided material from the professor. We believed the changes allow for a more comprehensive design to be implemented as this allows for a more efficient and practical database. As well, we decided to proceed with the MEAN (Mongo, Express, Angular, Node.js) Stack software bundle as opposed to the suggested alternatives of Ember and Firebase.

In our decision to update our Entity Class diagram, we focused on several areas of improvement that we could identify. The first being that the newly selected design removed any partial and transitive dependencies within the tables, thus eliminating any inconsistencies within the data (as learned in our Databases course, 3309). Another positive revision we recognized was due to the creation of separate tables for relational data that was shared between tables. By making this separation, we were able to remove any update anomalies that could erroneously skew the data. Lastly, the user table design (comprised of Patient, Administrator and Physiotherapist) in the given class diagram made use of more organized distribution of common attributes between all Self Start users. We saw value in taking this design over our previous one, as it made better use of storing information in proper tables within the database and allowed for easier access and insertion of data. Using this newly formed database schema, we are confident that we will be able to model the transferable data utilized by the Self Start system.

**Technical Decisions:**

Through our decision to utilize the MEAN Stack software bundle, we feel as though with our specific skills set we will be more capable of creating a final design that will not only meet but exceed the expectations of the client. Ember is a powerful framework that has incredible capabilities, but our team felt less confident in our ability to meet expectations using this framework. Our team has extensive development experience using the MEAN stack from our Web Technologies course from last semester as well as personal projects that we have completed on the side. In ensuring that our finished product is not only functional but eloquent, we will be employing ngbootstrap. This gives us a vast styling library to draw from, ensuring a tailored final product. We will be utilizing Angular 4 in the development of our Self Start web application. This decision was made solely based on the level of experience that we all have with the version, as it is the only one we have practical experience with.

# User GIT Branches

|  |  |  |
| --- | --- | --- |
| Person | Branch | Contributions |
| Andrew Black | <https://github.com/UWO-ECE-Software-Engineering/Loop-Solutions-Inc/tree/AndySemTwo> | * Added 3 routes * Revised schemas * Revision explanation |
| Samuel Mallabone | <https://github.com/UWO-ECE-Software-Engineering/Loop-Solutions-Inc/tree/SamSemTwo> | * Created 5 tables * Added 2 routes * Revision explanation |
| Robert Northmore | <https://github.com/UWO-ECE-Software-Engineering/Loop-Solutions-Inc/tree/MattSemTwo> | * Added 2 routes * Configured .gitignore |
| Stephanie Pereira | <https://github.com/UWO-ECE-Software-Engineering/Loop-Solutions-Inc/tree/steph2> | * Created 5 tables * Edited SDS document |
| Jak Terpak | <https://github.com/UWO-ECE-Software-Engineering/Loop-Solutions-Inc/tree/jterpakSem2> | * Created 5 tables * Wrote technical decision SDS section |
| Craig Cook | <https://github.com/UWO-ECE-Software-Engineering/Loop-Solutions-Inc/tree/ccook87_sem2> | * Created 4 tables * Revised all routes |

**Updated Contributions – Deliverable 3**

|  |  |  |
| --- | --- | --- |
| Person | Area of Work | Description |
| Andrew Black | Image upload | Created a frontend service to allow the user to upload and manage images on the site. Updated backend to be able to store images. |
| Andrew Black | Book Appointment Form | Created the book appointment and client information sheet forms. |
| Jak Terpak | Manage User accounts | Admin now has the ability to view all clients and physiotherapists and is able to manage both of them. The admin can now view the clients that are specific to a physiotherapist. |
| Jak Terpak | General Styling | General styling of the page such as an admin side bar and redesigned the accordion view. |
| Craig Cook | Book Appointment | Helped in creating book appointment and worked to create the client information sheet. Refer to Andrew’s Book Appointment Form for more detail. |
| Craig | Email Verification HTML | Designed the html behind the Email Verification email. |
| Robert Northmore | Book Appointment Calendar/Time Slot Selector | Designed the booking system for client’s to select times for appointments. It is the in the form of a grid for ease of use for the client. |
| Stephanie Pereira | Manage User Accounts | Supported Jak in creating the Manage User Accounts functionality. Refer to Jak’s Manage User Accounts row for full description of work. |
| Sam Mallabone | Client Sign Up | Created the client sign up page for when a client wants to register for the website. |
| Sam Mallabone | Patient List – Additional Sort Parameters | Changed the Patient list so that the physiotherapist can choose to sort the list based on other parameters such as first name or last name. Also allows the physiotherapist to sort the list in ascending or descending order. |
| Sam Mallabone | Email Verification | Added in email verification so that the user has to click on a link sent to their email to verify their account. |

**Schema Updates**

In this section, the updates made to the schemas are listed.

|  |  |  |
| --- | --- | --- |
| Table | Additions | Description |
| Exercises | -Changed Multimedia | Changed the multimedia field to properly allow for image querying. |
| Image | -Created Image Table | Created the image table to store images. |
| Patient | -Added reference to physiotherapist | Made this change so that a physiotherapist’s patients could be listed. |
| Temp | -Created the Temp table | Made a temp table to store a user’s information who is waiting to be verified. |
| Patient | -Added verified field | Verified field is a Boolean on whether the client has verified their account or not. |
| Patient | -Added in address field | Added in a field to store the address of a client. |

**Updated Contributions – Deliverable 4**

|  |  |  |
| --- | --- | --- |
| Person | Area of Work | Description |
| Stephanie Pereira | Physio Calendar | Worked on using a calendar for the user to see all their appointments. It has a month view, week view and day view |
| Craig Cook | Landing Pages | Worked on developing all the landing pages that a person will see upon coming to the sight. Created all the landing pages that unauthorized users can see. |
| Robert Northmore | Scheduler | Continued to work on the scheduler that the user will use for booking appointments. This will support the feature of setting availability and booking time off |
| Jak Terpak | Assessment Test | Reconfigured the assessment test functionality to add correct functionality for how an assessment test is related to a rehab plan |
| Jak Terpak | Complete Assessment Test | Added in the functionality for a new table being created when a client fills out an assessment test. It records the user’s answers and prompts physio revision |
| Andrew Black | Make Payment | Added in the functionality for the site to accept payments from a client. Different packages prompt different amounts to be paid |
| Andrew Black | Generate Patient Report and Visualize Treatment Results | Added the functionality for visualizing the results from completed assessment tests. Also added in the feature for the physio giving feedback to a client. |
| Andrew Black | Dashboards | Created the dashboards for the client, physiotherapist and admin. |
| Sam Mallabone | Generate Patient Report and Visualize Treatment Results | Added the functionality for generating the patient report which can then be saved, printed or emailed to the patient |
| Sam Mallabone | Authentication | Added in authentication for the site with an authenticated user getting a session token to use the site and its services. It stops unauthorized access of certain functionality |
| Sam Mallabone | Change Password | Added in functionality for all user’s to change their password as well as have their password reset if they have forgotten it |

**Schema Updates**

In this section, the updates made to the schemas that were made before deliverable 4 are listed.

|  |  |  |
| --- | --- | --- |
| Table | Additions | Description |
| Complete Assessment Test | Created the table | Created a table for completed assessment test’s which store the client’s answers and have a link to the client |
| Initial Intake | Created the table | Created the table initial intake which stores a user’s answers to the initial intake form |
| User Account | Added fields. | Added fields for date registered, last logged in, is disabled, reset request sent. These are all used as functionality in the site |
| Session | Created Table | Created a table to store the sessions that are currently active within the site |
|  |  |  |

**Major Functionality Added For Deliverable 4**

* Authenticate User
* Make Payment
* Change Password – Have your password be reset. Change your password yourself
* Update Account information
* Generate Patient Report and Visualize Treatment Results
* Dashboards and improved navigation