|  |
| --- |
| UAH Fit Vault Detailed Design |
| CPE 656/658 Software Studio |
| Timothy R. Wilkins  Whit J. Sisulak  Glen L. Riden  James J. Duggan IV |

11/13/2015

Table of Contents

[1. DETAILED DESIGN OVERVIEW i](#_Toc435182014)

[1.1 Controllers i](#_Toc435182015)

[1.1.1 SelectDataController i](#_Toc435182016)

[1.1.2 SelectActivityController ii](#_Toc435182017)

[1.2 Entities iii](#_Toc435182018)

[1.2.1 Activity iii](#_Toc435182019)

[1.3 Enumerations iv](#_Toc435182020)

[1.3.1 ActivityEnum iv](#_Toc435182021)

[1.4 Sequence Diagrams v](#_Toc435182022)

## DETAILED DESIGN OVERVIEW

### Controllers

#### SelectDataController

This class is responsible for selecting the data files for processing. This class provides the backend for the **SelectDataView**. See Figure 1 sequence diagram for this controller.

**<<Task>>**

**Attributes:**

* **Private Collection<int, String>** DataFiles
  + Dictionary containing the paths to the data files and a unique id.
* **Private String** DirectoryPath
  + Path to the directory containing the data files to be processed.

**Operations:**

* Sets and gets for all attributes.
* **Public ActionResult** btnFolderBrowseClick()
  + When the folder browser button is clicked a folder browser dialog object is created, a directory is selected by the user, and the DirectoryPath is set.
  + The results are returned to the **SelectDataView.**
* **Public ActionResult** DragAndDropFiles()
  + When the desired files are selected and dropped into the GUI window.
  + The results are returned to the **SelectDataView.**
* **Public ActionResult** btnClearClick()
  + When the clear button is clicked any stored data and file path(s) will be removed.
    - DirectoryPath variable reset
    - DataFiles collection emptied
    - The results are returned to the **SelectDataView.**
* **Private ActionResult** btnNextClick()
  + When the next button is clicked the file locations of the files in the DirectoryPath are added to the DataFiles collection.
  + The DataFiles collection is passed to the **SelectActivityController.**

#### SelectActivityController

This class is responsible for assigning an activity to a date and time portion of a data file. This class provides the backend for the **SelectActivityView**. See Figure 1 sequence diagram for this controller.

**<<Task>>**

**Attributes:**

* **Private Collection<Activity>** Activities
  + Activities is a collection used to hold all **Activity** objects.

**Operations:**

* Sets and gets for all attributes.
* **Public ActionResult** SelectFile()
  + A data file is selected (id).
  + The results are returned to the **SelectActivityView**
* **Public ActionResult** SelectActivity()
  + An activity is selected.
  + The results are returned to the **SelectActivityView**
* **Public ActionResult** AddDateRange()
  + A date and time range is selected.
  + The results are returned to the **SelectActivityView**
* **Public ActionResult** AddActivity()
  + An activity object is created and assigned the values from the selections
    - File id
    - Activity
    - Date time range
  + The results are returned to the **SelectActivityView**
* **Public ActionResult** ProcessData()
  + Iterates through the data files.
    - Calls the Validate method to validate each data file
      * If a file is invalid it is not process and any associated activates are dropped.
      * If a file is valid the UploadDataFile method is called and passed the data file object
        + Any activities associated with the data file is uploaded via the UploadDataActivity method.
  + When processing is complete the results will be returned to the **SelectActivityView**
* **Public ActionResult** Validate(**File file**)
  + Returns if the file is valid or invalid.
* **Public ActionResult** UploadDataFile(**File file**)
  + This method uploads the data file into the database.

* **Public ActionResult** UploadActivity(**Activity activity**)
  + This method uploads the activity into the database.

### Entities

#### Activity

This class contains data associated with an activity tied to a date and time portion of a data file.

**<<Entity>>**

**Attributes:**

* **Private int** fileId
  + The id of the file to be assigned an activity.
* **Private DateTime** TimeStamp
  + The date and time of the activity.
* **Private ActivityEnum** DataActivity
  + The activity of the data.

**Operations:**

* Sets and gets for all attributes.

The **Activity** class is used by the **ActivitySelectionController**.

### Enumerations

#### ActivityEnum

This is the enumeration that contains user roles within the system. It is an **Enumeration**.

**<<Enumeration>>**

**Attributes:**

* Running
* Walking
* Sleeping
* Sitting
* Standing

### Sequence Diagrams

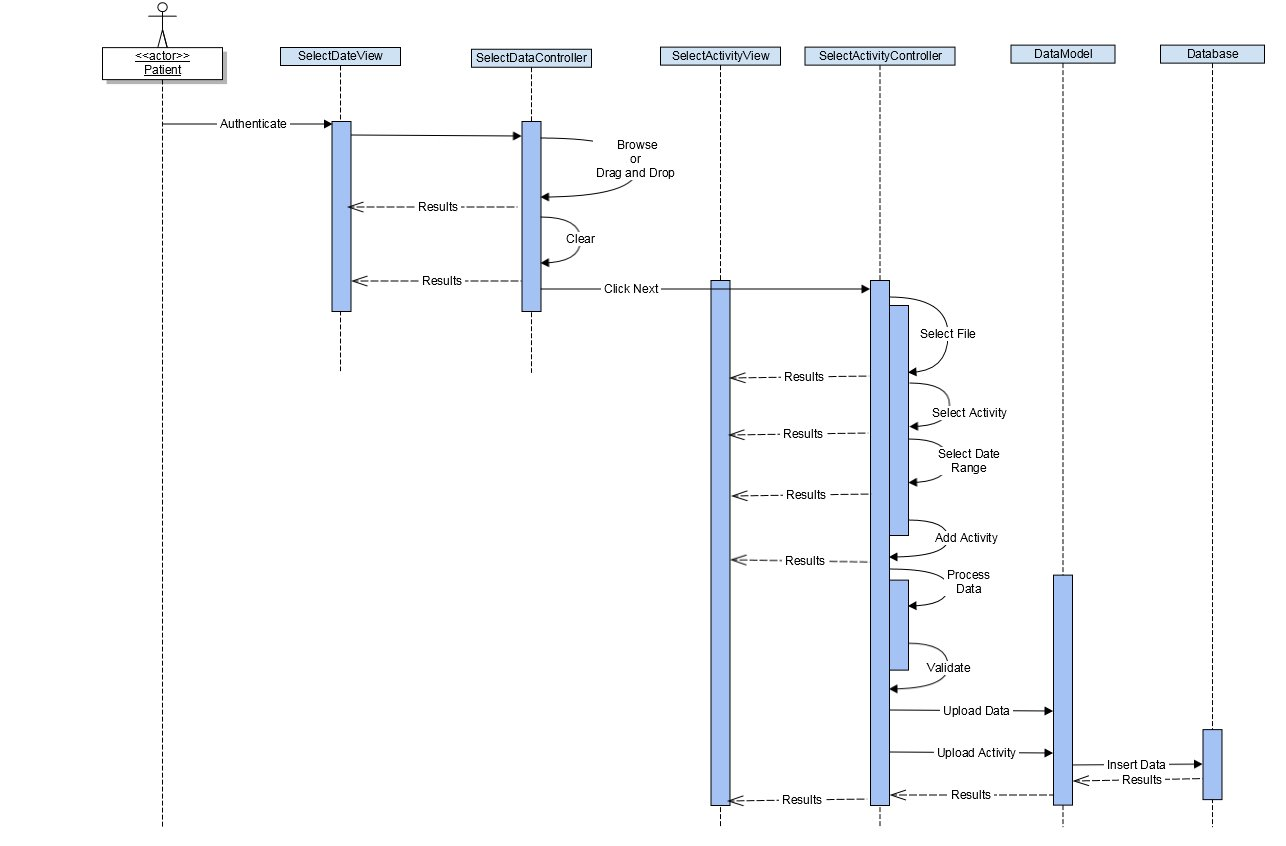


Figure : Data Processing Sequence Diagram