

Crazy Code Surgeons



Business Logistics Solutions
1900 Commerce St, Tacoma, WA 98402
(253) 692-4000
godofhand.github.io/Code-Surgeons
CodeSurgeons@u.washington.edu

Design Document

Document prepared for:
Global Business Logistics
Freight Forwarding Service
1030 Logistics Ave,
Tacoma, Washington 98402

Table of Contents

1. INTRODUCTION.....	pg.2
2. RATIONALE SUMMARY.....	pg.3
3. CLASS DIAGRAM.....	pg.5
4. USER STORY SEQUENCE DIAGRAM.....	pg.6
5. SYSTEM STARTUP SEQUENCE DIAGRAM.....	pg.11

INTRODUCTION

This is our work in the second phase of the program based on interviews with Director Arden Heale. The program is a fully standalone Language Library which helps to reduce drafting time of responses to Request for Proposal (RFP) from customers. A language library is a database of titles or words and their associated texts which are relevant to a company's information. RFP often has special format or structure with special needs. In most case, they ask a lot of similar or same information. Such questions may be general information about the company, financial viability and brief history. Our program dramatically improves the efficiency to respond to such RFP by quickly supplying users with accurate pre-created information to match those titles and words.

After our prototypes were approved by Ms. Heale, we started to design the language library focusing on important features such as searching, creating, and viewing. Searching is the most important feature in our application. By searching for keywords or titles, user can view the description of titles and text content. We also provide the ability of add custom titles and texts into the library. In addition, user can view the entire library as necessary. Our design implements the Model and View pattern. Implementing this way allow us to effectively couple the classes to avoid tight relationship and to improve cohesion.

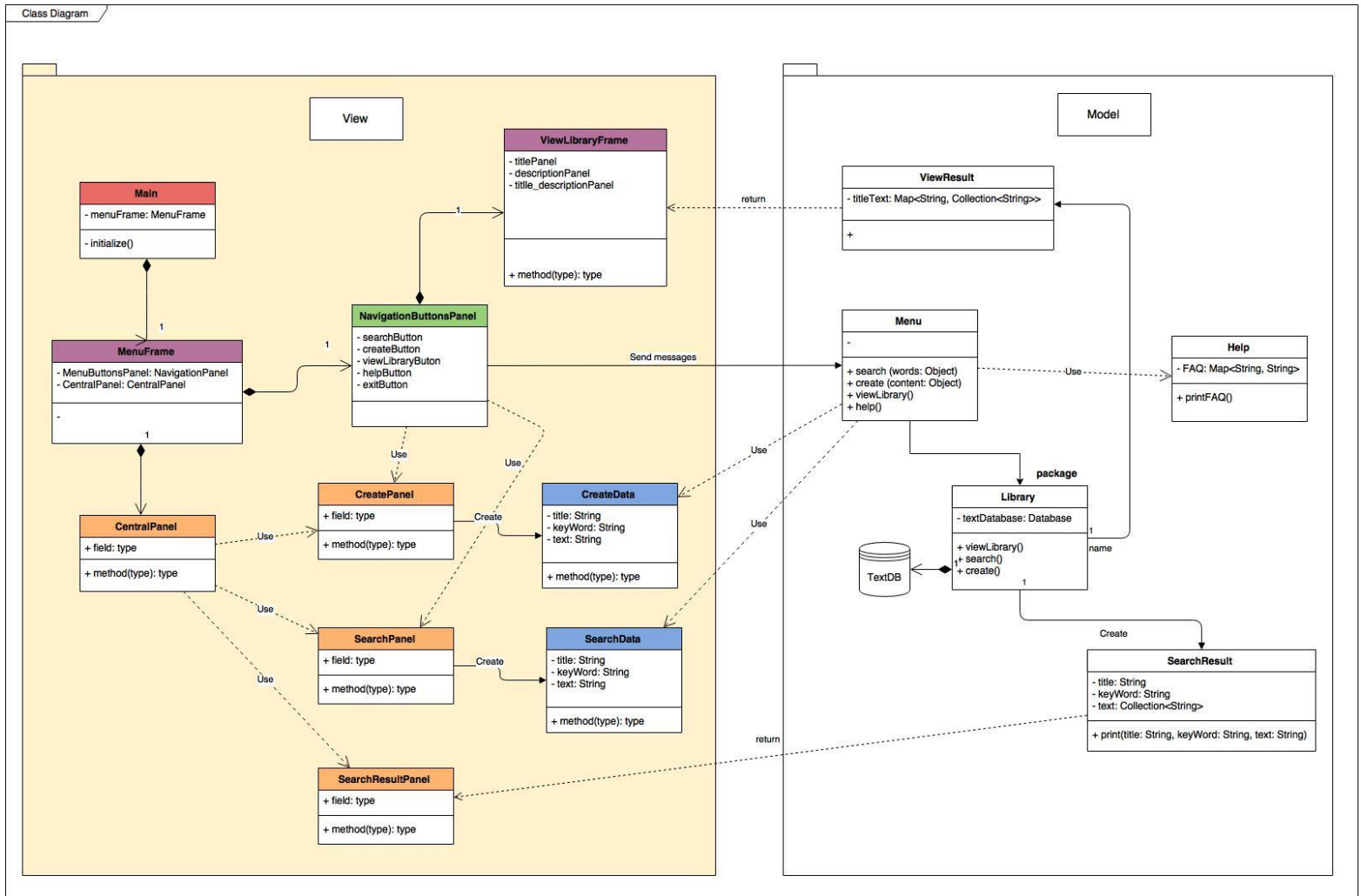
RATIONALE SUMMARY

Our design reflects the Model and View pattern. The user interacts through classes in the View. The View collaborates with classes in the Model to operate on data input by user. We chose this design pattern because it is simple to use and allow loose coupling to be implemented easier without sacrificing cohesion and reusability.

The Model of our design is straightforward. The primary communication between the Model and the View is handled by Menu class. Although we are aware this class may potentially be a god class in term of communication, the design is necessary to limit access from View to the Library which contains critical and confidential data. Menu is where we can perform check for data and command messages from View. If user clicks help button in View, Menu will call Help class to display its list of FAQ. Other supported operations in Menu are handled down to Library class. It has a database to store all entries and three essential behaviors, viewLibrary, search, and create. Since the Library has a database, it is easier to perform searching and creating entry inside the Library. For search operation, the Library creates SearchResult class that contains the searched data. This class is returned to the View in order to it to update the search results that appear on theSearchResultPanel. For view library operation, the Library creates a new ViewResult that contains a map of titles and their associated list of texts. This class is then returned to the ViewLibraryFrame to display.

All the components in View are exclusively contained by one class, MenuFrame. Based on our paper prototype, we decided for MenuFrame to hold a NavigationButtonPanel, consisted of five essential functional buttons, and a central panel that displays the corresponding functional panel, SearchPanel, SearchResultPanel and CreatePanel, in response to the buttons. The button panel is static and its buttons do not change their positions over the session of the application. Implementing only two big components in this way allows us to send messages from the button panel and central panel separately to the Model. This composition is geared toward the decision that at any point, the user can navigate to desired functional panels by clicking the buttons. In addition to functionality of the NavigationButtonPanel, we added a ViewLibraryFrame class to display the entire the library should the user clicks view button. The data that user enters will be encapsulated in two data class, CreateData and SearchData, and then are passed to the Model. Note that in our design, the MenuFrame actually does not contain any behaviors that response to user's actions. Rather, the behaviors are part of NavigationButtonPanel, SearchPanel, and CreatePanel to avoid creating a god class.

CLASS DIAGRAM

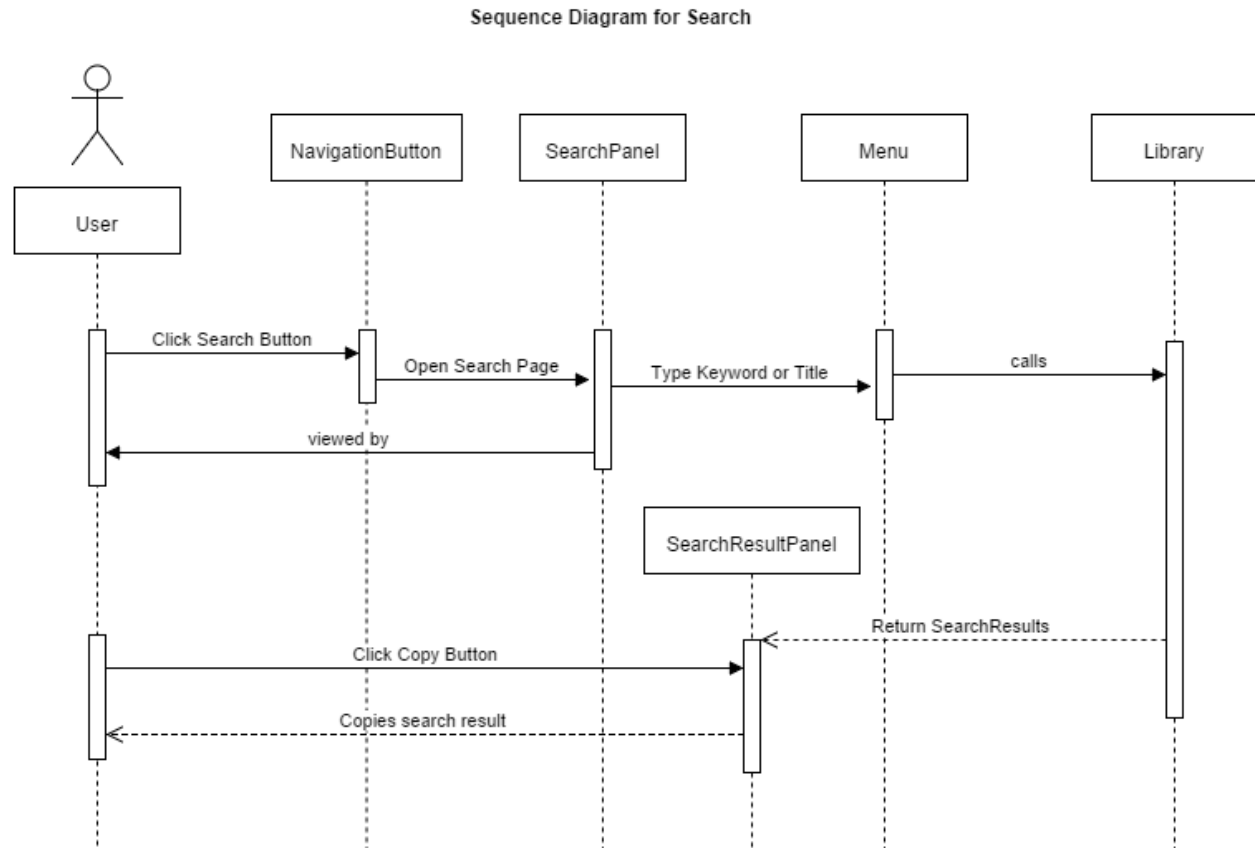


USER STORY SEQUENCE DIAGRAM

The sequence diagram (SD) specifies the time and control aspects of a system.

Typically, SD is used to analyze only the more complex events. Events are actions between the objects in the project. They can also transmit data. A user story can be one of the event during one execution of a program. A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime user story in a graphical manner, so that it would be easier for the user to follow the sequence of activities and flow of data.

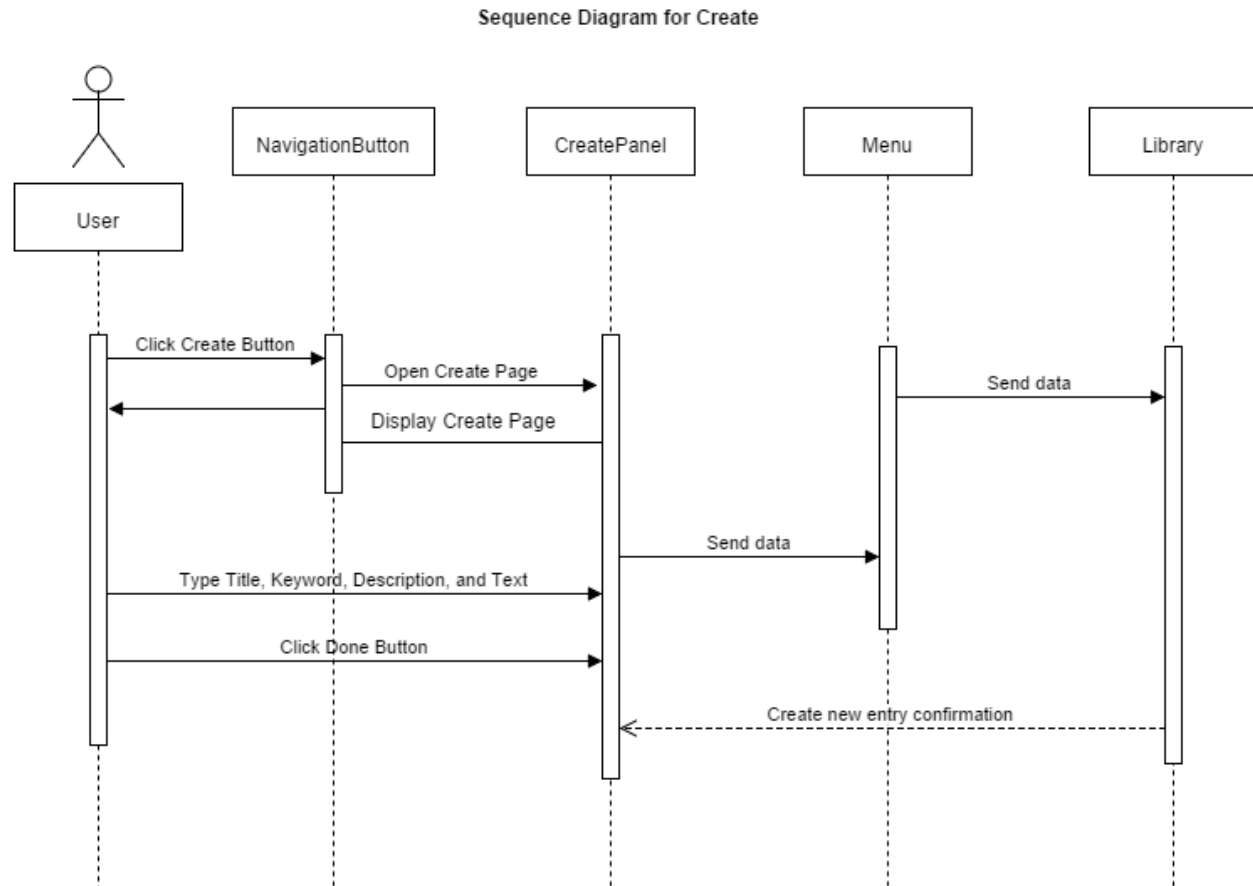
User story 1: As a user I want to search for texts using title and keywords so that I can add them to my proposal.



Brief Description: This sequence diagram is initiated by users to view search results. The initial requirement is doing the search by Keyword or by Title and having the results.

Initial Step-By-Step Description: There is no initial use for the user to login for using this sequence diagram. 1. User clicks Search Button. 2. Search Panel displays “Search” window which consists search form. 3. User enters Keyword or Title. 4. System calls library by Keyword or by Title. 5. System displays the results. 6. User clicks the Copy button in front of results if he/she wants to copy the search results. 7. Search Result Panel copies the search results. 8. User clicks cancel button to close this page.

User Story 2: As a user I want to create blocks of texts associated with a title so that I can use them later.

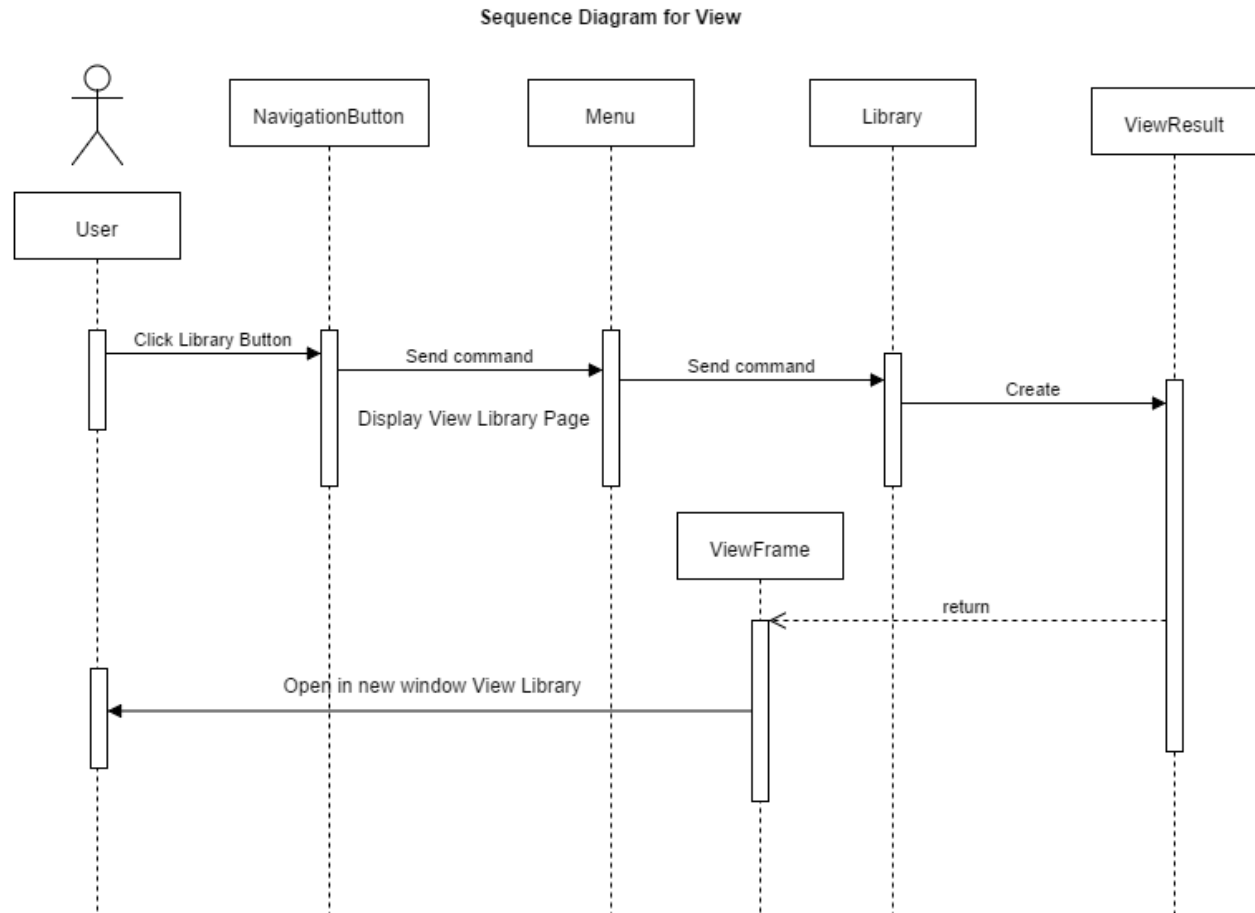


Brief Description: This sequence diagram is initiated by users to create new data to the library.

The initial requirement is doing the create Title, Keyword, Description, and Text.

Initial Step-By-Step Description: There is no initial use for the user to login for using this sequence diagram. 1. User clicks Create Button. 2. Create Panel displays “Create” window which consists create form. 3. User enters Keyword, Description, and Text. 4. User clicks Done button. 5. System sends data to the library by Title, by Keyword, by Description, and by Text. 6. System shows new entry confirmation. 7. User clicks Main Menu button to go back to the Main Menu page.

User Story 3: As a user I want to view the complete library of keywords and their associated texts so that I don't have to create new ones.

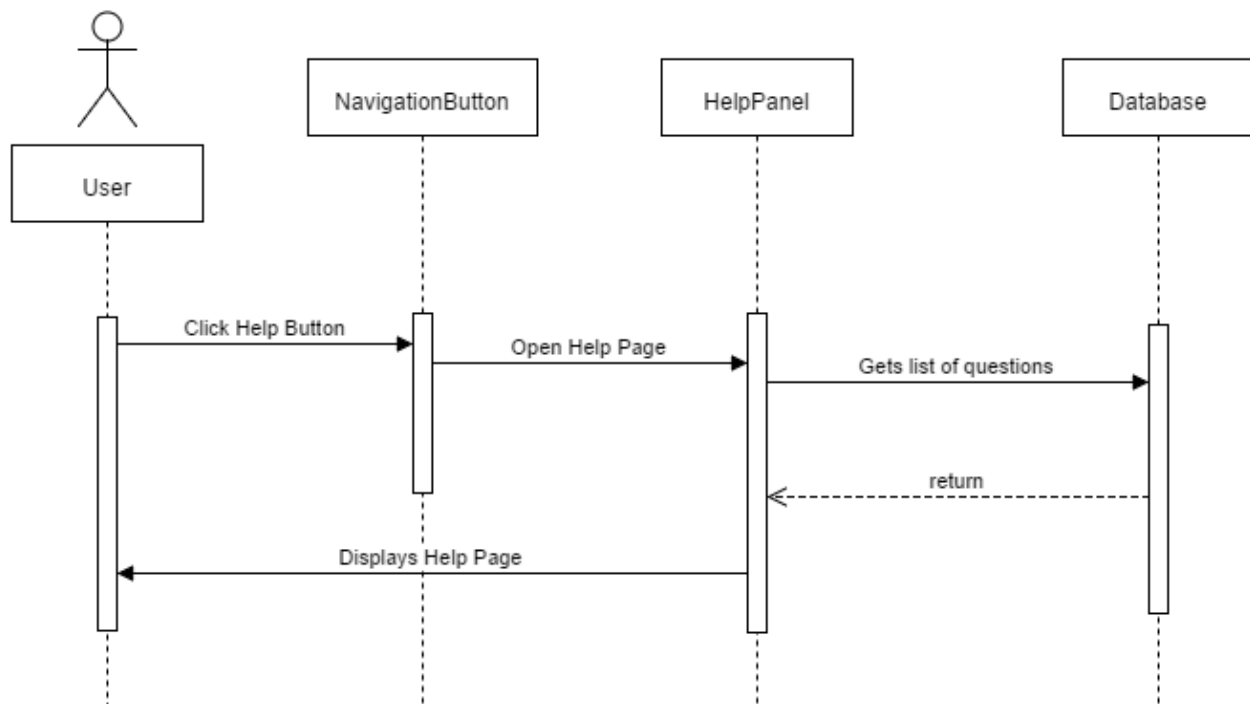


Brief Description: This sequence diagram is initiated by users to View data in the library. The initial requirement is doing the View Title, Keyword, Description, and Text.

Initial Step-By-Step Description: There is no initial use for the user to login for using this sequence diagram. 1. User clicks Library Button. 2. System sends command to library 3. Main Menu separately displays “View Library” window which consists data. 4. System shows results from the library. 7. User clicks done button to close this page.

Recommended User Story from Usability Test: As a user I want to get help so that I can use the application more efficiently.

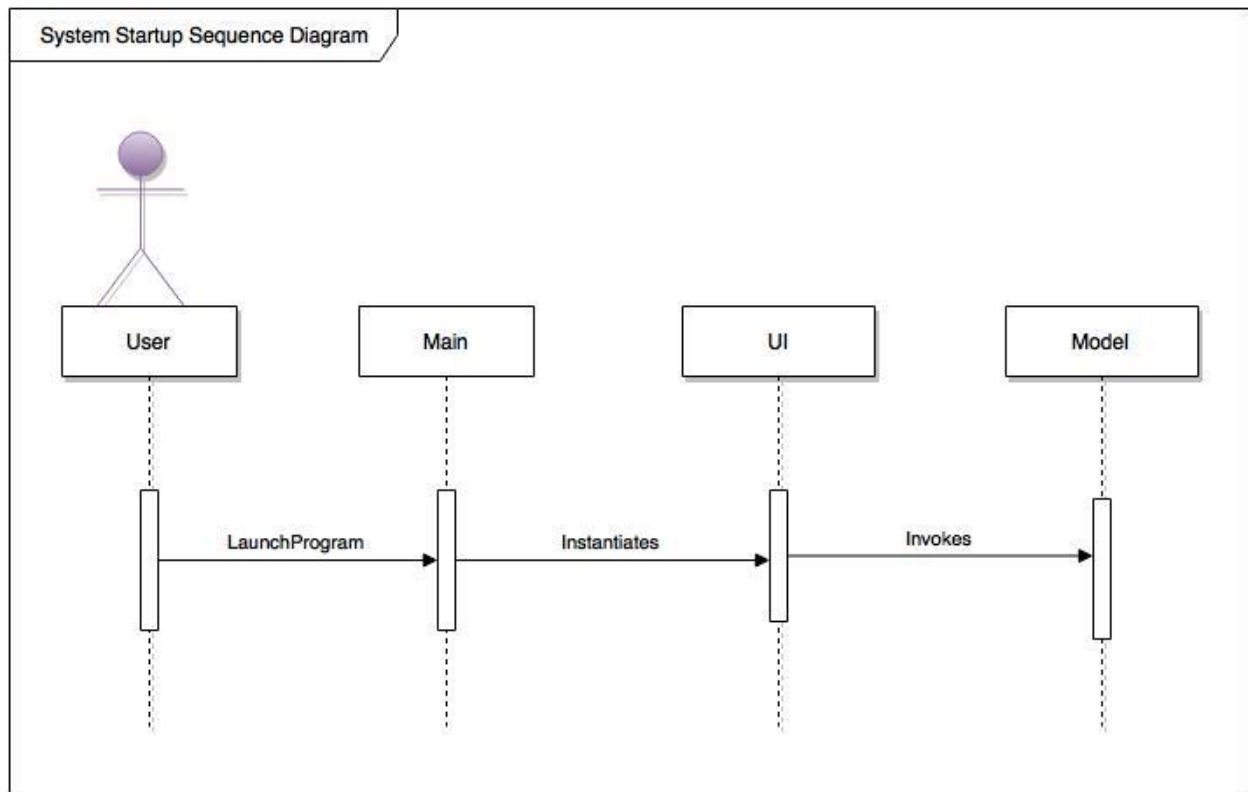
Sequence Diagram for Help



Brief Description: This sequence diagram is initiated by users to get Help information stored in the database.

Initial Step-By-Step Description: There is no initial use for the user to login for using this sequence diagram. 1. User clicks Help Button. 2. Help Panel displays “Help” window which consists help information. 3. System displays list of questions. 4. User clicks Main Menu button to go back to the Main Menu page.

SYSTEM STARTUP SEQUENCE DIAGRAM



Brief Description: This sequence diagram is initiated by users to start the program.

Initial Step-By-Step Description: When the program is run by user, main class is executed first and instantiates a GUI object which is the MenuFrame in the class diagram. Any other classes that are related to MenuFrame are instantiated such as NavifgationButton and CentralPanel. Then the GUI invokes methods in the Model under the user's command.

----- END OF DOCUMENT -----