Assignment 3: Use Cases

Team members: ______Talal Jawaid______________ ________Shiwei Fang____________ ________Amrit Singh_________ ___________Sergio Zavala_____________

Learning Objectives

· Make interaction designs and represent them using use case diagrams and use case descriptions.

Success Criteria

· Be able to read and write use case diagrams and use case descriptions.

Resources

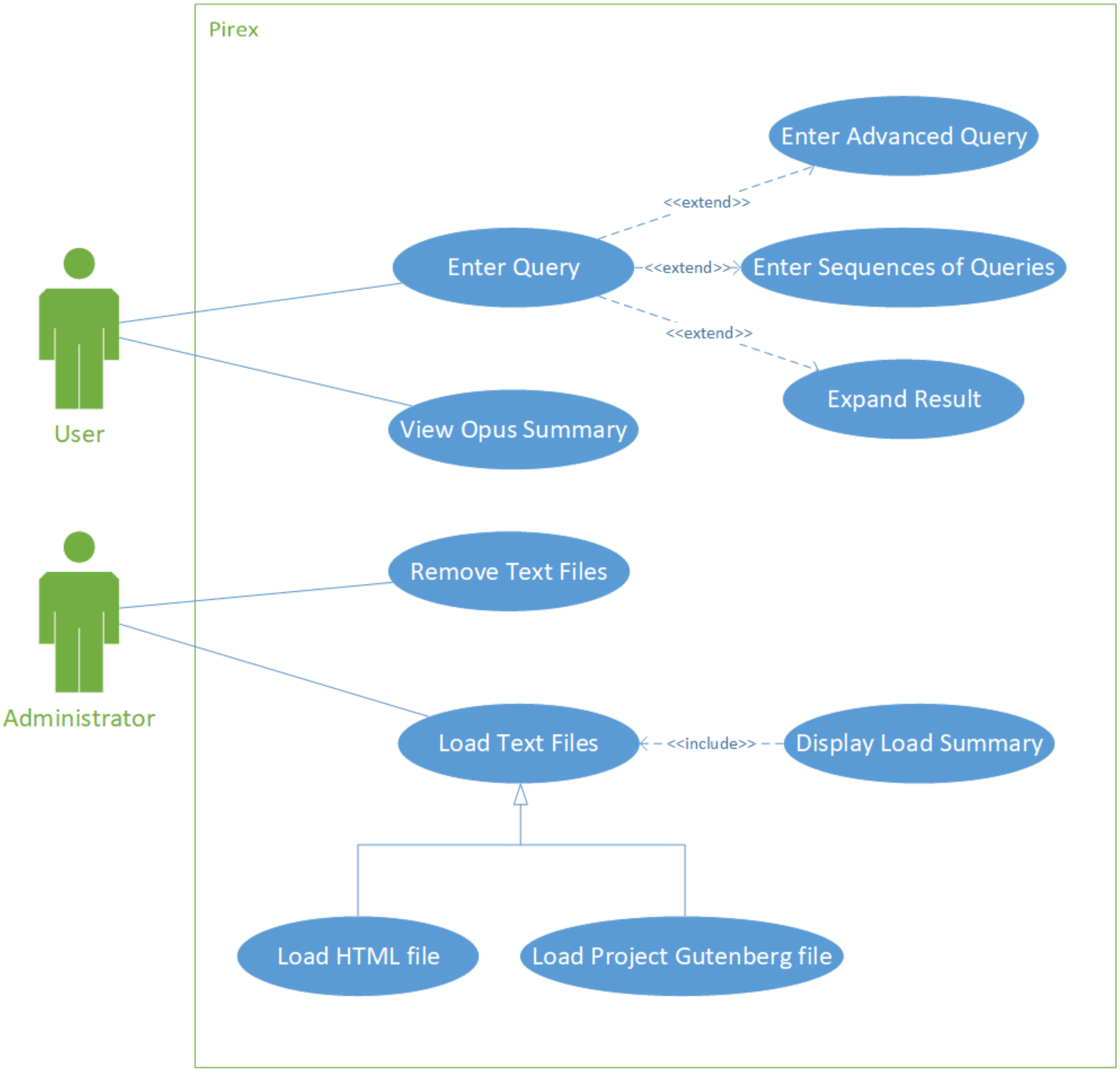
- Use Cases (ebook chapter)
- *Week04_Requirements.pptx* (in Week4 Module, Canvas)
- *Interaction Design* (IDD_v1.pdf)
- *Project SRS* (SRS_v1-1.pdf).

Problems (Deliverable)

1. Based on the project SRS, what are the actors in the Pirex system? (5 pts)

The primary actors are the administrators and the users. There are no secondary actors.

2. Make a high-level use case diagram for Pirex (using UML Use Case template in Visio), which shall capture all use cases, actors, external systems, etc. (20 pts)


Pirex
User
Administrator
Enter Advanced Query
Enter Query
Enter Sequences of Queries
Expand Result
View Opus Summary
<<extend>>
<<extend>>
<<extend>>
Remove Text Files
Load Text Files
<<include>>
Display Load Summary
Load HTML file
Load Project Gutenberg file

3. Each team member choose a use case (i.e., an oval) in your diagram and make a use case description for it using the template which is given below. For a team with 3 members, at least 3 use case descriptions shall be delivered. (60 pts)

- *Use Case Name*: To identify the use case
- *Actors*: The agents participating in the use case
- *Stakeholders and Needs*: What this use case does to meet stakeholder needs
- *Preconditions*: What must be true before this use case begins
- *Post conditions*: What will be true when this use case ends
- *Trigger*: The event that causes this use case to begin
- *Basic Flow*: The steps in a typical successful instance of this use case
- *Extensions*: The steps in alternative instances of this use case occurring either because of variations in the normal ow or because of errors.

UML Diagram Individual

Shiwei Fang

· *Use Case Name*: **Enter and Search Query**

· *Actors*: Users (principal actors)

· *Stakeholders and Needs*: As the core function of this software provides searching capabilities of the stored texts in the system, this use case fulfills the basic functional requirements of users who intend to use this software. Basic queries can be entered to retrieve documents; advanced queries may be utilized using various operators; queries are displayed in short form (but can be expanded to long form if desired); users may enter a sequence of queries for several queries in a row. Users are afforded full access to texts that fulfill their search parameters.

· *Preconditions*: There must exist texts to search from.

· *Post conditions*: A short form summary of each relevant text is displayed.

· *Trigger*: A search query is entered.

· *Basic Flow*: The user opens enters a query they wish to search for. They may also enter an advanced query with the relevant operators if they wish. Once the user presses the return key, a search is executed for the relevant parameters. A list of all relevant texts are displayed in a list in a short form summary. The user may then choose any text on the list in particular and expand the entire document in long form display. After the user is done with their search, they may choose to continue with a subsequent query, repeating the process as many times as they would like.

Sergio Zavala

· *Use Case Name*: **View Opus Summary**

· *Actors*: The agents participating in the use case

- The actor will be the user.

· *Stakeholders and Needs*: What this use case does to meet stakeholder needs

- This use case summarizes the stored documents. This benefits the user because it searches for the enter query and displays all the different files that has the same files.

· *Preconditions*: What must be true before this use case begins

- The user must enter a query that the user wants to search for, file selection, and load file selected.

· *Post conditions*: What will be true when this use case ends

- It will search for the documents tab and it will load documents tab for user.

· *Trigger*: The event that causes this use case to begin

- The event that causes this use case to begin is the searching for documents.

· *Basic Flow*: The steps in a typical successful instance of this use case

- Blank query, search results, file selection, load file selected, and load summary

· *Extensions*: The steps in alternative instances of this use case occurring either because of variations in the normal ow or because of errors.

  - There is no extensions for this use case.

Amrit Singh

· *Use Case Name*: **Remove Text Files**

· *Actors*: The agents participating in the use case

- The actor in this is just the administrator. Only the administrator has the permission to remove text files.

· *Stakeholders and Needs*: What this use case does to meet stakeholder needs

- Unnecessary files are removed for the users benefit. Lessens the amount of documents a user would need to search through.

· *Preconditions*: What must be true before this use case begins

- There needs to be unnecessary file(s) in the system.

· *Post conditions*: What will be true when this use case ends

- There will be no unnecessary files in the system.

· *Trigger*: The event that causes this use case to begin

- There are useless/unnecessary files in the system.

· *Basic Flow*: The steps in a typical successful instance of this use case

- Load all files, sort unnecessary files from necessary files, remove unnecessary files

· *Extensions*: The steps in alternative instances of this use case occurring either because of variations in the normal flow or because of errors.

- There are no extensions needed for the Remove Text Files use case.

Talal Jawaid

· ***Use Case Name***: **Load Text Files**

· ***Actors***: The agents participating in the use case

The administrators are the ones participating in this use case.

· ***Stakeholders and Needs***: What this use case does to meet stakeholder needs

This meets the users need of allowing text to be loaded to be able to search through.

Users need text to be able to search through and this use case makes that text loaded into Pirex

· ***Preconditions***: What must be true before this use case begins

The current primary actor must be an administrator

- ***Post conditions***: What will be true when this use case ends

Pirex will have one more text file loaded into it.

- ***Trigger***: The event that causes this use case to begin

The administrator clicking the load files tab in Pirex will cause this use case to begin

- ***Basic Flow***: The steps in a typical successful instance of this use case

The administrator will click the **Load Documents** tab. They will then click **Browse** and select a text file. If the file is not a Project Gutenberg file, they will change the default **Text File Type.** If it's a **Project Gutenburg** file, **Pirex** will autofill the Title and Author fields. Otherwise **Pirex** will require the administrator to fill the Title and Author fields. The administrator will then click on **Process.** At this point, the results of the loading process will be displayed in the text box at the bottom.

- ***Extensions***: The steps in alternative instances of this use case occurring either because of variations in the normal ow or because of errors.

In the event that Pirex is capable of loading HTML files, it will have to parse the data in the HTML file so that it can be properly searched through.