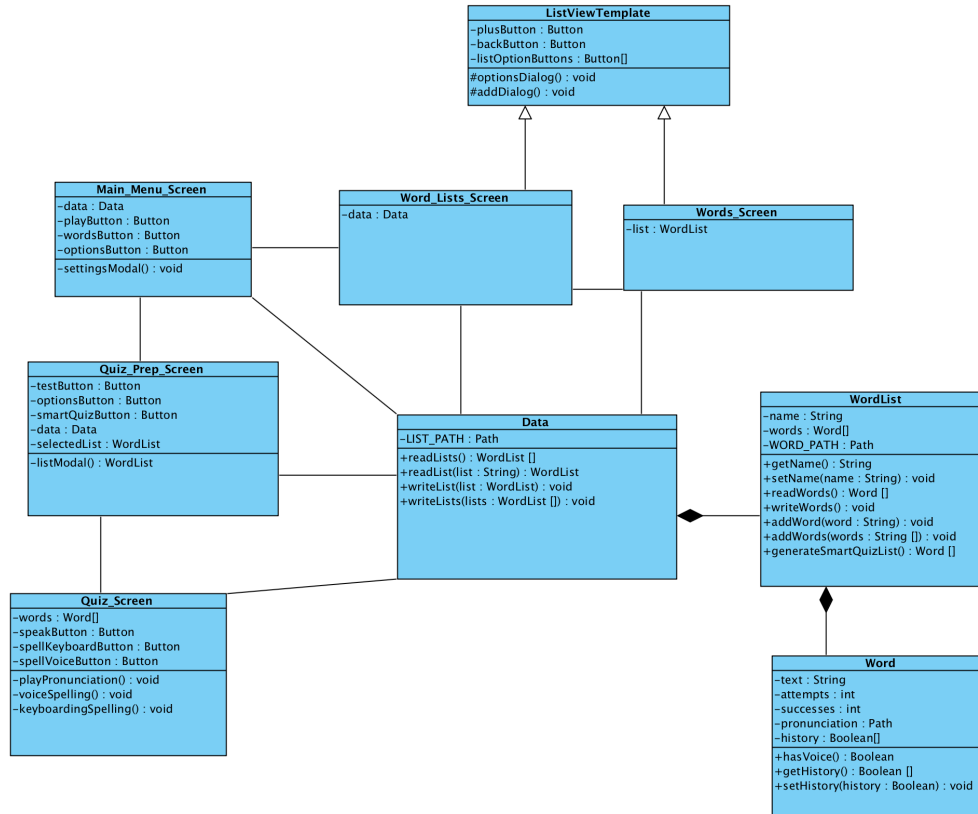# Bee Spelled

Naomi Perry, Thomas Lewallen, Jack Dwyer

Problem Definition

- There is a need for an efficient way to study for a spelling bee or spelling test
- The most common way of doing so involves being tested by a friend/family member
- Current applications lack desired functionality (saying word out loud, voice recognition, asking specific words, etc.)
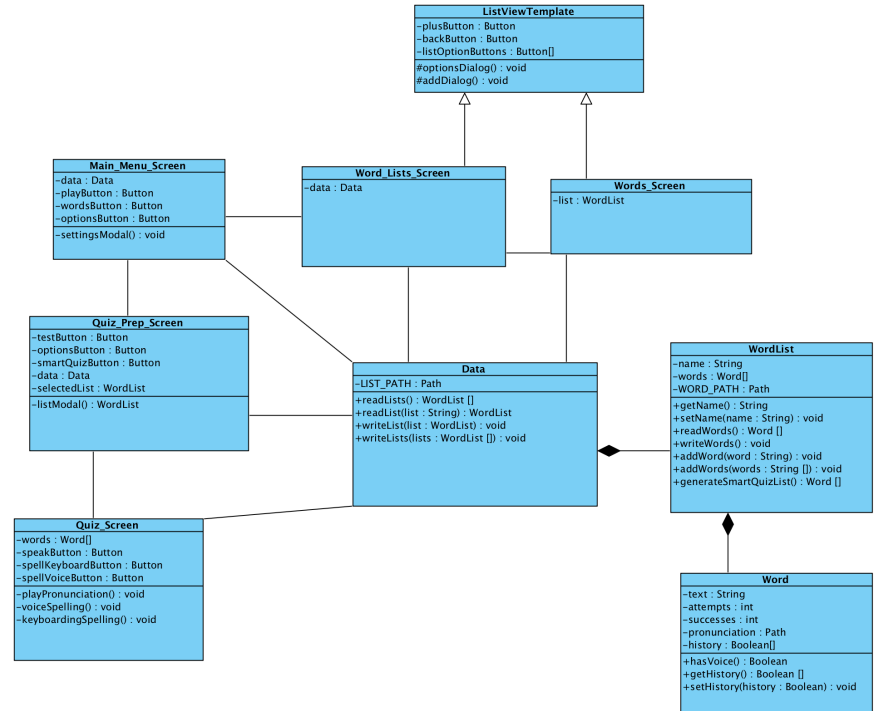
Scope of System

- The system will allow users to create their own lists of words by typing and pasting words.
- Users will be able to edit lists and individual words.
- Users will also be able to hear the words.
- The system will recognize the user both speaking and typing their answers.
- Users will be able to save lists for future use.
- The system will test in an intellectual way: keeping track of problem words, and percentages of correctness.
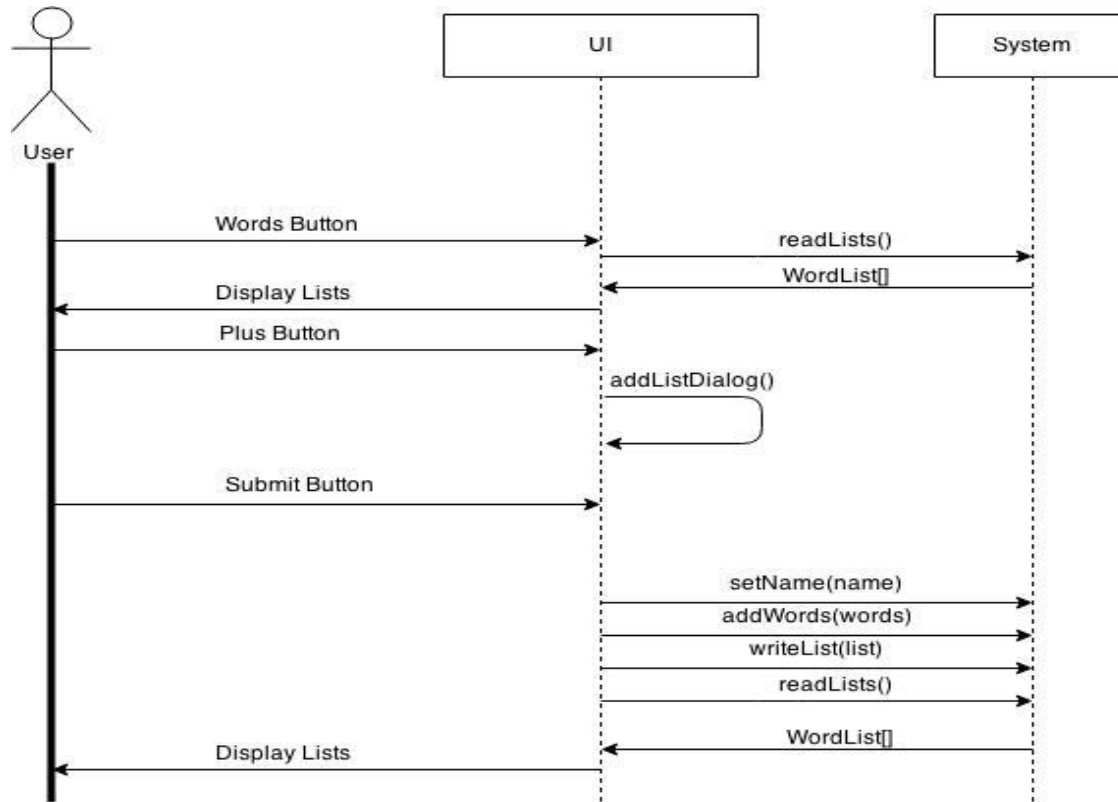
Class Diagram

ListViewTemplate
- –plusButton : Button
- –backButton : Button
- –listOptionButtons : Button[]

#optionsDialog() : void
#addDialog() : void

Main_Menu_Screen
- –data : Data
- –playButton : Button
- –wordsButton : Button
- –optionsButton : Button

–settingsModal() : void

Word_Lists_Screen
- –data : Data

Words_Screen
- –list : WordList

Quiz_Prep_Screen
- –testButton : Button
- –optionsButton : Button
- –smartQuizButton : Button
- –data : Data
- –selectedList : WordList

–listModal() : WordList

Data
- –LIST_PATH : Path

+readLists() : WordList []
+readList(list : String) : WordList
+writeList(list : WordList) : void
+writeLists(lists : WordList []) : void

WordList
- –name : String
- –words : Word[]
- –WORD_PATH : Path

+getName() : String
+setName(name : String) : void
+readWords() : Word []
+writeWords() : void
+addWord(word : String) : void
+addWords(words : String []) : void
+generateSmartQuizList() : Word []

Quiz_Screen
- –words : Word[]
- –speakButton : Button
- –spellKeyboardButton : Button
- –spellVoiceButton : Button

–playPronunciation() : void
–voiceSpelling() : void
–keyboardingSpelling() : void

Word
- –text : String
- –attempts : int
- –successes : int
- –pronunciation : Path
- –history : Boolean[]

+hasVoice() : Boolean
+getHistory() : Boolean []
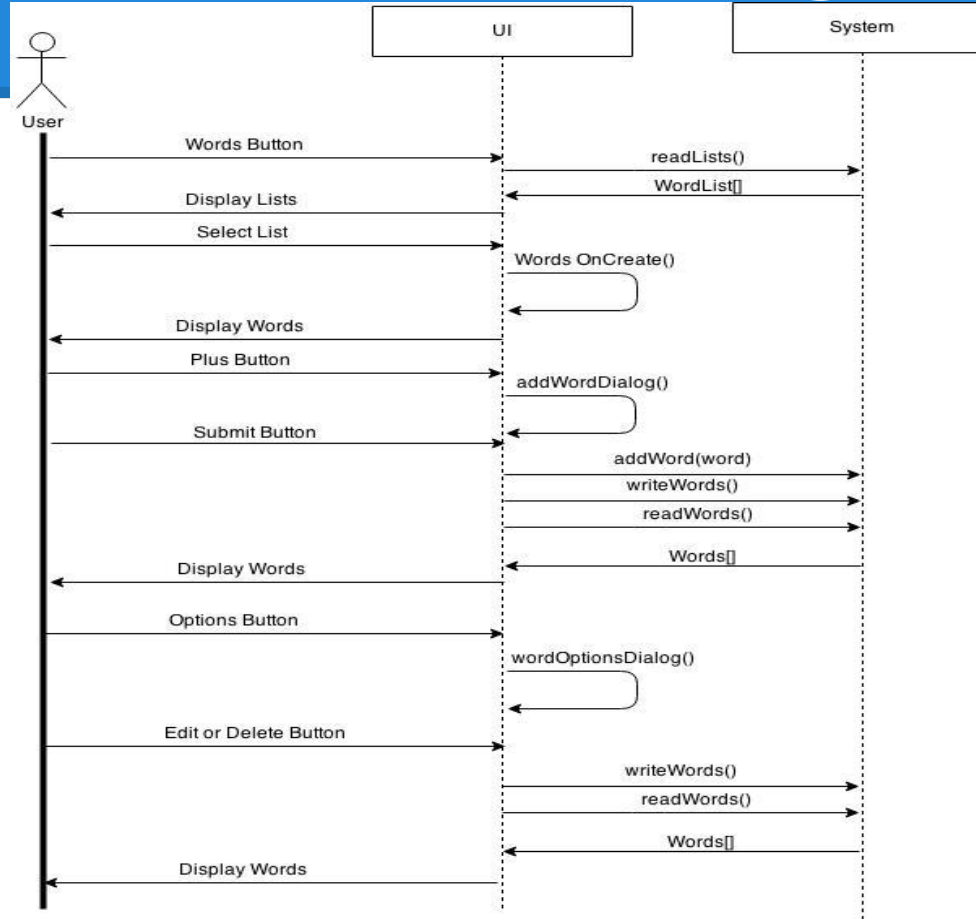+setHistory(history : Boolean) : void

Class Specifics…

- All data is held in files - Serialized objects
- Main_Menu_Screen will create a Data object, pass it to other screens
- Data object will read in Lists and Lists will read in Words
- Word_List_Screen and Words_Screen will be ListView screen which will list all the WordLists and Words in a WordList respectively.
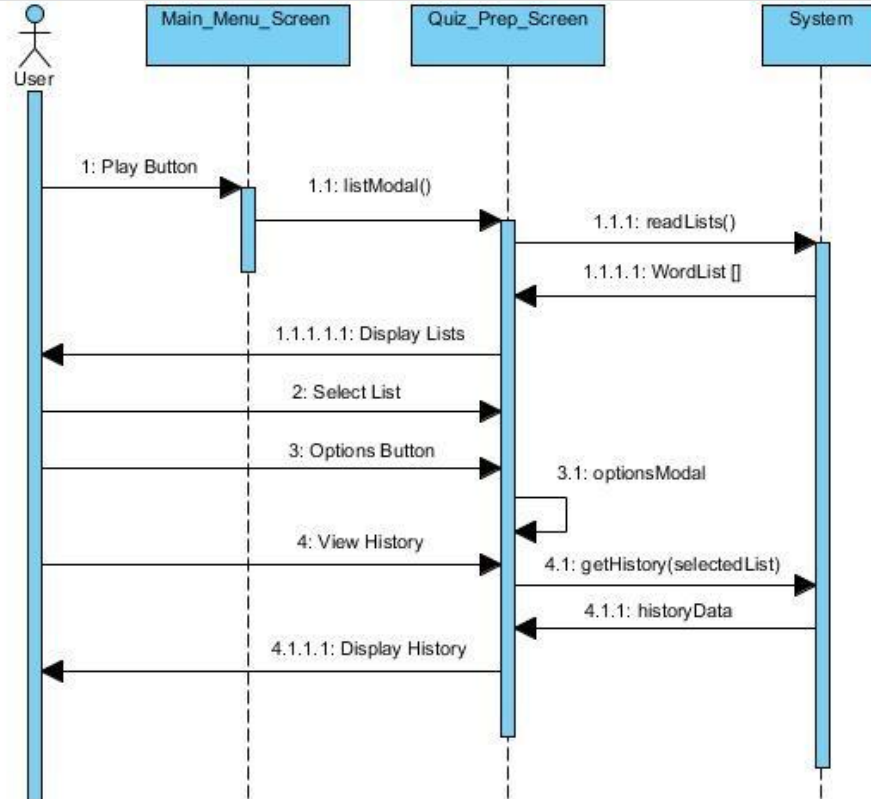- Changes will be written to files when exiting Words_Screen and Word_Lists_Screen and when a quiz ends.
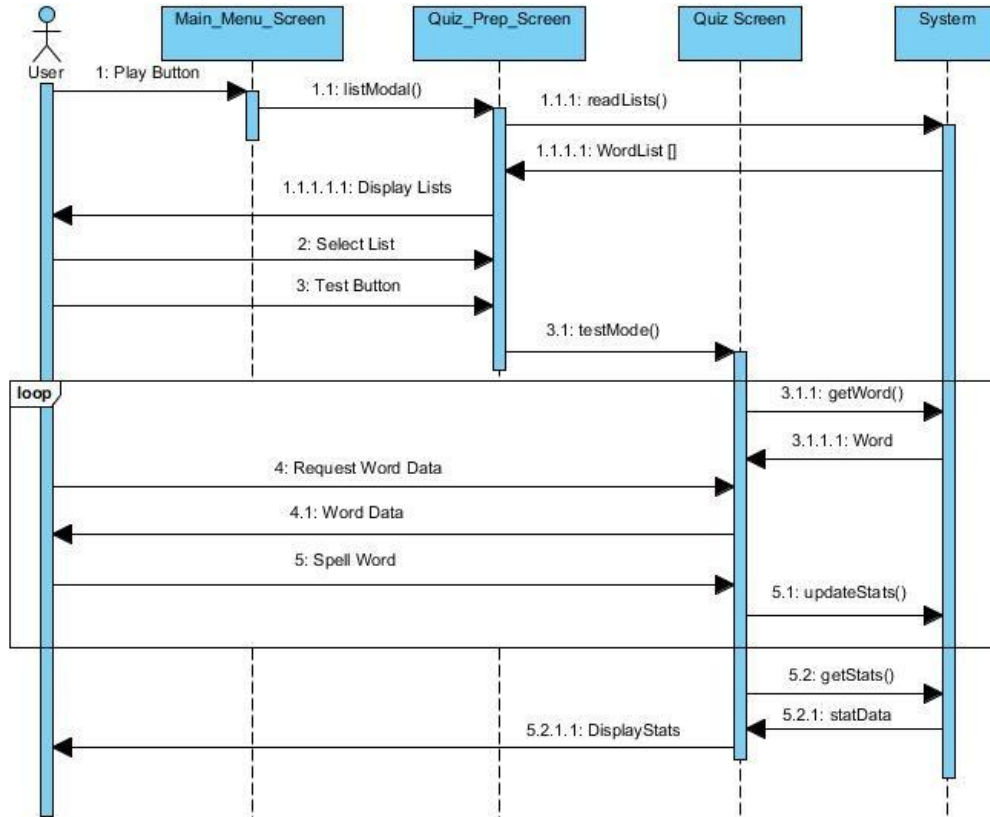
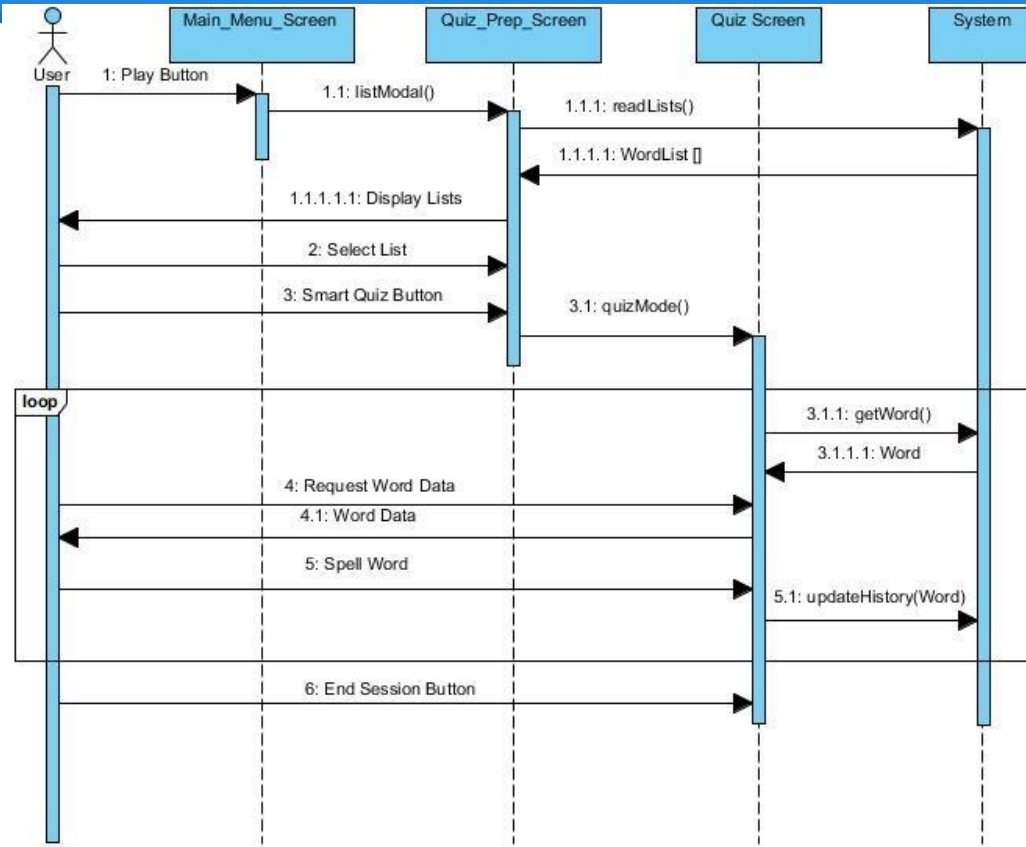# Create List Sequence Diagram

Edit List Sequence Diagram

View Statistics Sequence Diagram

Test Sequence Diagram

Smart Quiz Sequence Diagram

Changes Since Requirements

- Removed Merriam-Webster API functionality
- Added to design to save word-specific data, i.e. spelling accuracy and spelling history
- Changed all data storage to Serializable Objects that will be written to files

Perspective

- Scope has shrunk a little bit
- Chose simple solutions, but left room for expansion to include some of the original ideas
- Spelling via voice works as expected, however, may require some fine tuning