On Call
Software Design Specification

A. Archer (aa), K. Hosaka (kah), A. Huque (ash), L. Jim (lmj), M. Terry (mht)
3-08-2020 - v2.03

Table of Contents

1. SDS Revision History

Date	Author	Description
2-12-2020	lmj	Created the initial document.
2-12-2020	lmj	Created *On Call* Architecture Diagram Updated References and Acknowledgments sections.
2-13-2020	lmj	Wrote section 2 and 3.
2-13-2020	ash	Proofreading and edited sections 2 and 3.
2-14-2020	lmj	Proofreading and minor edits.
2-14-2020	aa	Proofreading, minor comments and edits.
2-14-2020	mht	Final edits and comments before submission.
2-15-2020	lmj	Wrote primary functions in section 4. Updated sections 2 and 3 including software architecture diagram.
2-16-2020	lmj	Wrote the rest of section 4, excluding models.
2-18-2020	lmj	Updated section 4.
2-19-2020	aa	Created and added section 5 diagrams. Updated Acknowledgements.
2-19-2020	ash	Formatting
2-19-2020	lmj	Created and added section 4 models. Updated References and Acknowledgements. Fixed formatting b/c of new figures added.
2-19-2020	aa	Minor diagram change.
2-20-2020	mht	Final edits before submission
3-01-2020	kah	Proofreading, fixed formatting, editing for "Good Writing", adding undo functionality in earlier sections.
3-02-2020	kah	Proofreading, fixed formatting, editing for "Good Writing", adding undo and reset functionality in middle sections.
3-03-2020	kah	Changing tense throughout SDS.
3-04-2020	kah	Proofreading, fixed formatting, editing for "Good Writing", adding undo and reset functionality in final sections. Adding Prof. Hornof's feedback throughout SDS.
3-04-2020	aa	Suggestions and feedback
3-05-2020	kah	Proofreading
3-05-2020	ash	Proofreading for accuracy, edited captions, section 3, and 4.2
3-05-2020	lmj	Updated section 4 diagrams. Updated sections based on suggestions and feedback.
3-06-2020	aa	section 4.4 suggestions for rewrite.
3-06-2020	lmj	Updated sections based on suggestions and feedback
3-07-2020	ash	Added deleting individual RA to all sections pertaining to raPreferences
3-07-2020	lmj	Proofreading, small edits, and formatting.
3-08-2020	lmj	Updated diagrams and expanded captions.
3-08-2020	mht	Updated section 4.3

2. System Overview

- The terms user and coordinator are used interchangeably in this SDS as the target user group are RA coordinators.
- Upon starting *On Call*, the user is greeted by the home view built by the On Call Viewer module. On Call Viewer is the only module that users directly interact with. This module allows the user to interact indirectly with the RA Preferences module and the Shift Assignments module.
- The user can upload CSV files containing weekday and weekend preferences for each resident assistant, or RA, through On Call Viewer. The file is sent to RA Preferences which will read, parse, and save RA information.
- RA Preferences receives settings specified by the user. This includes a list of RAs who cannot share shifts, a gold star RA, and the method to break tied preferences.
- The Weekday Scheduler and Weekend Scheduler modules process the information saved by RA Preferences. Weekday Scheduler uses the weekday requirements and given RA preferences to schedule all of the weekday shifts for the term. Weekend Scheduler uses the weekend requirements and given RA preferences to schedule all of the weekend shifts for the term.
- Because the weekday and weekend requirements differ significantly, these processes happen separately in their own modules. Both schedulers send their respective shift information to the Shift Assignments module.
- Shift Assignments takes the information received from the scheduler modules and formats the information into an actual schedule, saving the generated schedule in the system.
- On Call Viewer allows the user to export a CSV file containing the on call schedule that it receives from Shift Assignments. It also allows the user to export a text file containing a summary report of the schedule in Shift Assignments.
- On Call Viewer allows the user to view and edit information saved by RA Preferences as well as the schedule created by Shift Assignments.
- On Call Viewer allows functionality for users to undo an update that was made to RA Preferences and return to a previous state of the preferences. Users can also undo an update made to the generated schedule and return to a previous state of the schedule.
- On Call Viewer allows the user to reset RA Preferences so there are not RAs saved in the system. Additionally, Shift Assignments can also be reset to clear the saved schedule from the system.

3. Software Architecture

3.1. Components

- On Call Viewer - Provides the user a graphical interface to use. This is the only module that users interact with directly.
- RA Preferences - Provides functionality to import and update RA weekday and weekend preferences, choose settings (gold star RA, tiebreakers, and incompatible RAs), undo updates to RA preferences, and reset RA preferences.
- Weekday Scheduler - Provides functionality to assign RAs to every weekday shift according to weekday requirements.
- Weekend Scheduler - Provides functionaltiy to assign RAs to every weekend shift according to weekend requirements.
- Shift Assignments - Provides functionality to generate shift assignments, update shifts, undo updates, export the schedule, export a summary of shifts, and reset the schedule.

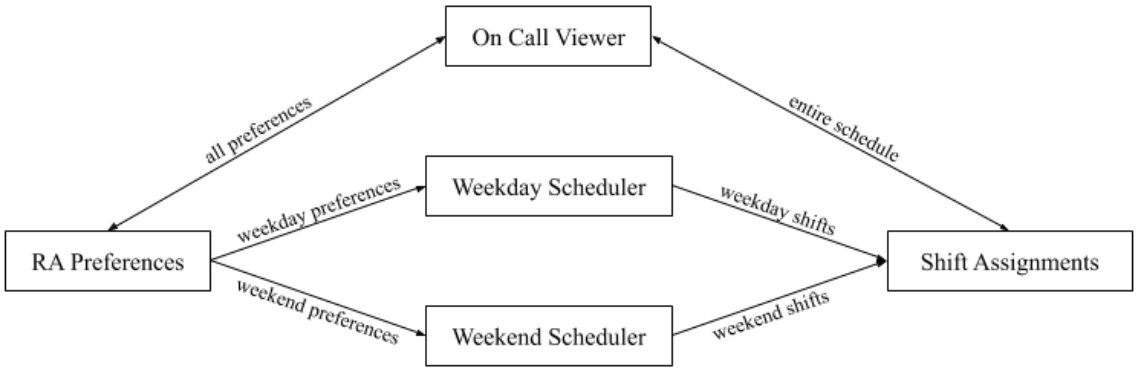

Figure 1. *On Call* Architecture Diagram. The boxes represent components.
The arrows indicate the flow of information sharing between modules.
The arrow labels indicate what information flows between the modules.

3.2. Component Interaction

- On Call Viewer and RA Preferences Interaction
    - On Call Viewer provides RA Preferences the name of the file being imported and RA Preferences returns the current RA information to display on the interface.
    - On Call Viewer provides functionality to update the saved RA information without another file import and can trigger RA Preferences to undo updates made to the RA preferences.
    - On Call Viewer sends the settings specified by the RA coordinator to RA preferences, which includes a list of RAs who cannot share shifts, a gold star RA, and the method to break tied preferences.

    - On Call Viewer provides RA Preferences with the name of a specific RA to permanently delete from the entire system.
    - On Call Viewer can trigger RA Preferences to reset, deleting all of the RAs from the system.
- RA Preferences and Weekday/Weekend Scheduler Interaction
    - RA Preferences provides Weekday Scheduler with each RA's weekday preferences and the settings information.
    - RA Preferences provides Weekend Scheduler with each RA's weekend preferences and the settings information.
- Weekday/Weekend Scheduler and Shift Assignments Interaction
    - Both Weekday Scheduler and Weekend Scheduler provide Shift Assignments with information about which RA fills which shift.
- Shift Assignments and On Call Viewer Interaction
    - Shift Assignments provides the schedule information to On Call Viewer or exports the schedule to a location specified by On Call Viewer.
    - Shift Assignments provides On Call Viewer with a way to update individual shifts to different RAs.
    - Shift Assignments can be triggered by On Call Viewer to undo an update made to an individual shift.
    - Shift Assignments can delete the schedule currently saved in the system when triggered by On Call Viewer to do so.
    - Shift Assignments exports a summary of shift assignments when triggered by On Call Viewer.

3.3. Design Rationale

- The On Call Viewer module sends and receives information with RA Preferences and Shift Assignments because the user needs a way to import/export, update, undo, reset, and view RA and schedule information through the user interface. The user also needs a way to provide RA Preferences with the selected settings related to preferences (such as a list of RAs who cannot share shifts, a gold star RA, and the method to break tied preferences).
- On Call Viewer is its own module because splitting the user interface functionality between RA Preferences and Shift Assignments would make those modules tightly coupled. By having On Call Viewer, there is strong intramodule cohesion and loose intermodule coupling between On Call Viewer, RA Preferences, and Shift Assignments.
- The RA Preferences module processes the information the user provides on RA weekday and weekend preferences. The information does not go directly from On Call Viewer to Weekday Scheduler and Weekend Scheduler because both modules require the information to be contained in a data structure and formatted in a specific way. The RA Preferences module prevents both of the schedulers from needing to process RA information from a CSV file and maintain the information together. RA Preferences handles any and all changes to the saved RA information, so there is no confusion where the functionality to update, undo, or reset the information is.

- The Weekday Scheduler module provides Shift Assignments the information on weekday shifts and the Weekend Scheduler module provides Shift Assignments the information on weekend shifts. This is done because both the weekday and weekend shifts need to be interwoven into one schedule. By having Shift Assignments as a separate module to handle combining the weekday and weekend shifts, Weekday Scheduler and Weekend Scheduler maintain loose coupling as they do not directly rely on each other to create a single schedule.
- Weekday Scheduler and Weekend Scheduler are separate modules because the requirements and the algorithms for scheduling RAs on the weekdays versus the weekends differs significantly. The separation of these two scheduling algorithms strengthens intramodule cohesion. Weekday Scheduler and Weekend Scheduler maintain loose coupling as they do not directly rely on each other to create a single schedule.
- The Shift Assignments module combines the information provided by Weekday Scheduler and Weekend Scheduler into one complete schedule. By having a single module maintain this information, On Call Viewer only needs to interact with one module when requesting,updating, undoing, or reseting the schedule information. This allows On Call Viewer to have loose coupling.

4. Software Modules

4.1. On Call Viewer

4.1.1. Primary Function

The On Call Viewer module provides the coordinator with a graphical user interface, or GUI. Through the GUI, the user can import, view, update, undo updates, and reset RA preferences. They can also select schedule generation settings by providing a list of RAs who cannot share the same shift, giving a gold star to one RA, and choosing a method to break a tie in preferences. Through the GUI, the coordinator can also generate, view, update, undo updates, reset the schedule, and export a schedule.

- User Interface:
  Using Tkinter, the GUI windows are created. Tkinter's label widget is used to display RA preferences as well as the schedule within the GUI. The GUI's buttons, such as import and export, are created using Tkinter's button widget. Additionally, Tkinter's filedialog module is used to accomplish importing and exporting files. Tkinter's combobox widget is used to allow the coordinator to update information in the system. The combobox widget is also used to select the RA that gets the gold star and select the tiebreaker method. Finally, Tkinter's combobox widget is also used to indicate which RAs cannot share a shift.

4.1.2. Interface Specification

- On Call Viewer interacts with RA Preferences.
  - When the user clicks a button to import a file, On Call Viewer passes the file name to a function in RA Preferences which processes the file. This file can either be new RAs with their preferences or updated information on existing RAs.
  - When the user requests to view the RA preferences in the system, On Call Viewer uses the dictionary RA Preferences modifies.
  - If the coordinator changes a preference using the user interface, the name of the RA, which preference was changed, and what the new preference is gets passed to RA Preferences to update the dictionary. On Call Viewer effectively has read, and not write, access to the dictionary.
  - When the coordinator selects a list of RAs who cannot share the same shift, gives a gold star to one RA, or chooses a method to break a preference tie, this information gets sent to RA Preferences to be saved in the preferences dictionary.
  - If the user clicks a button to undo the most recent change made to an RA's preferences, On Call Viewer triggers RA Preferences to undo the change and re-update the preferences.
  - If an RA is specified for deletion, On Call Viewer calls on RA Preferences to remove the specific RA from the dictionary.
  - When it is time to delete all of the RAs from the system to prepare for a new year, a button click will trigger RA Preferences to reset the dictionary of saved preferences back to empty.

- On Call Viewer also interacts with Shift Assignments.
    - When the coordinator requests a new schedule to get generated, On Call Viewer notifies Shift Assignments which triggers the weekday and weekend schedulers to run.
    - To display the schedule to the coordinator in the user interface, On Call Viewer accesses the dictionary Shift Assignments modifies. Since Shift Assignments maintains this dictionary, On Call Viewer effectively has read only access to it.
    - When the user updates a shift in the schedule, On Call Viewer passes Shift Assignments the information necessary to modify the dictionary.
    - If the coordinator wants to undo a change made to the schedule, On Call Viewer will trigger Shift Assignments to undo the most recent change.
    - On Call Viewer passes the name of a file to Shift Assignments which exports a CSV file of the schedule.
    - On Call Viewer passes the name of a file to Shift Assignments which exports a text file of a summary of the shift assignments by RA.
    - On Call Viewer can trigger Shift Assignments to reset the schedule if the user wants to remove the schedule saved in the system.

Models:

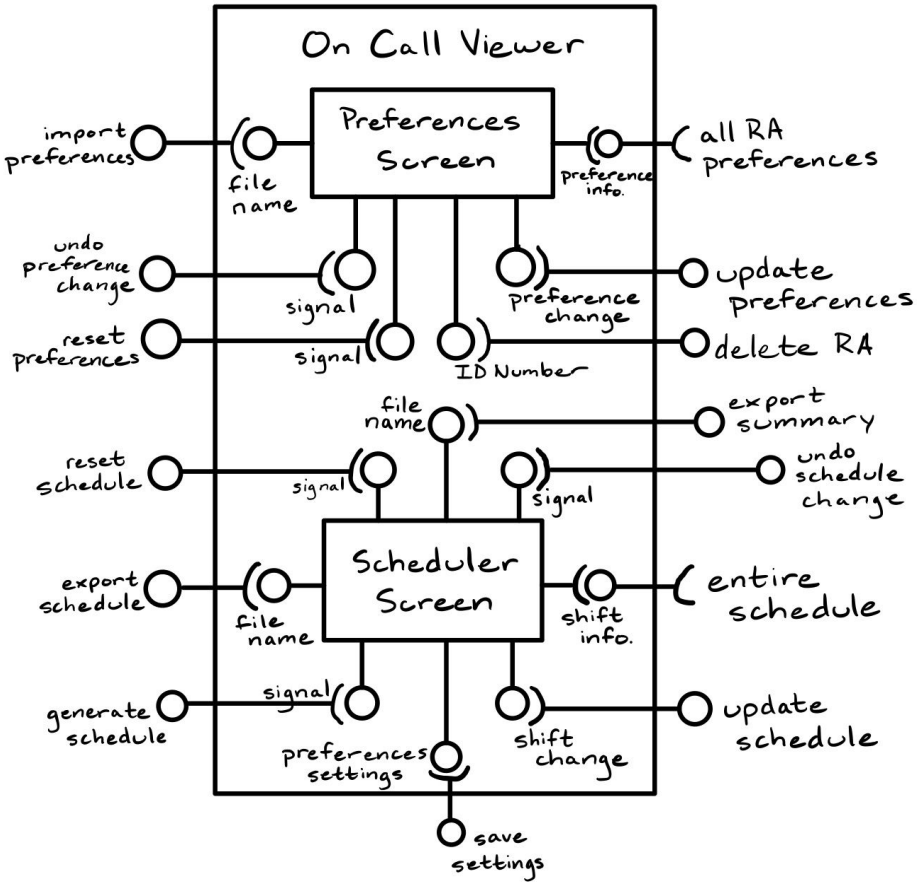

Figure 2. On Call Viewer UML Component Diagram.
The Preferences screen sends a file name to import preferences, signals to undo preference changes and reset preferences, ID numbers to delete RAs, and preference change information to update preferences.
The Preferences screen receives the preference information for all of the RAs.

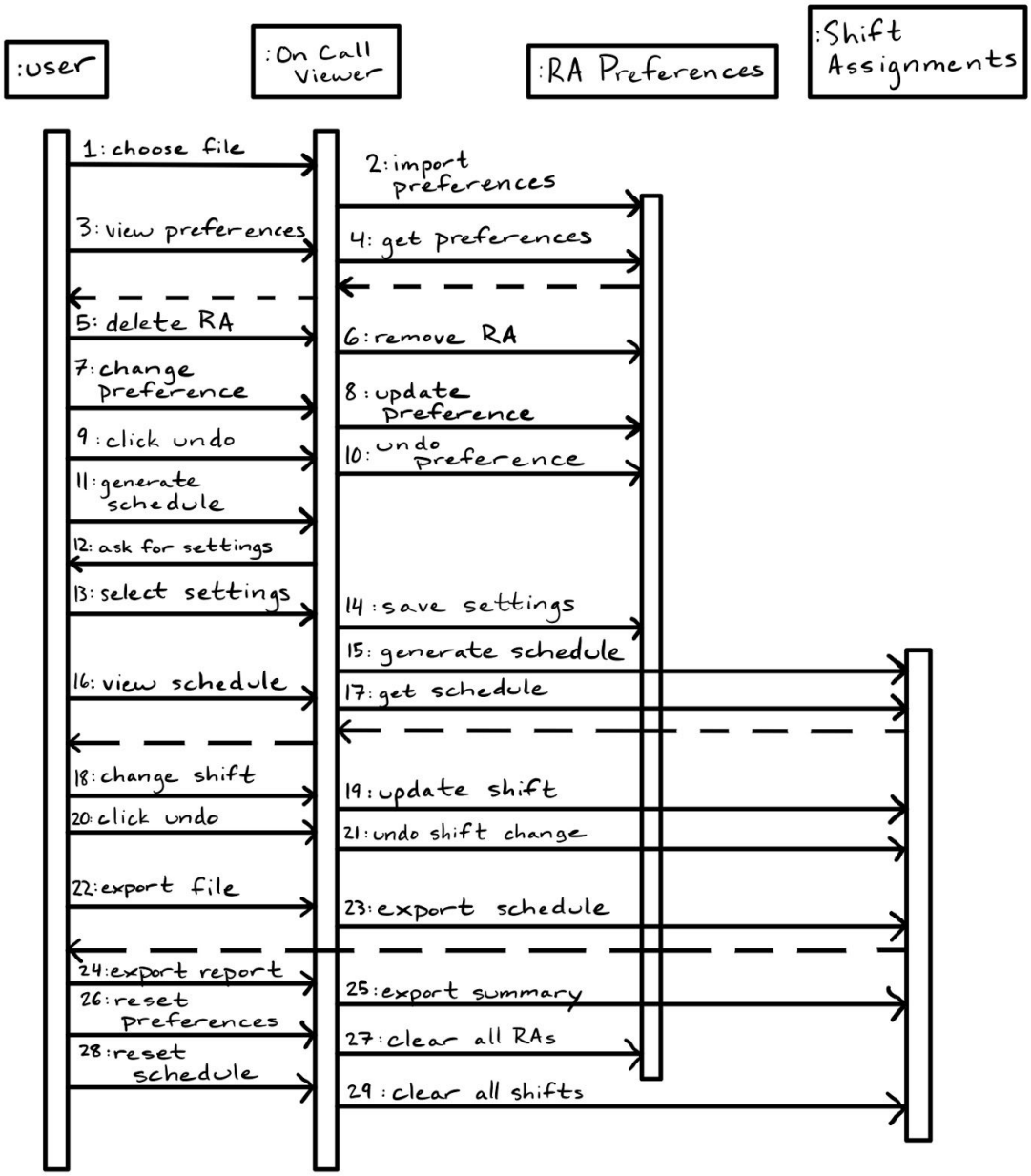

Figure 3. On Call Viewer UML Sequence Diagram.

First, the user chooses a file to import and it gets processed by RA Preferences. Then the user can view preferences through On Call Viewer, which gets the information from RA Preferences. The user can then select an RA to delete which gets removed from RA Preferences. The user can also change a preference which updates RA Preferences. The user can also undo this change. The user can then generate a schedule, which requires settings before actually generating. After generating the schedule, it can be viewed through On Call Viewer which gets the schedule from Shift Assignments. Shifts can then be updated, and those changes can be undone. The user can also export the schedule as well as a summary. Finally, the user can reset both the RA Preferences and Shift Assignments.

4.1.3. Design Rationale

- All of the graphical user interface functions are contained within this module. If another module contained interface functions, then it would need access to at least the root window. By having the GUI functions in only one module, the coupling is loosened.
- On Call Viewer relies on other modules to process files and modify saved information. This allows On Call Viewer to maintain strong cohesion as it only handles GUI functions and nothing else.
- The coupling is slightly tightened between modules by having On Call Viewer rely on RA Preferences and Shift Assignments, but On Call Viewer does not constantly rely on these modules. This means coupling is still fairly loose.

4.2. RA Preferences

4.2.1. Primary Function

The RA Preferences module imports, updates, undoes updates, deletes RAs, and resets the information on RA preferences.

- Import All Preferences:
    - One way the coordinator can import RA preferences is by having all of the RA preferences in one file with each line containing the preferences for an RA.
    - After receiving the name of the CSV file, each line is read, and the name of the RA along with their preferences are saved in a dictionary. This dictionary is written to a Python file, so the information is saved between sessions.
- Import Individual Preferences:
    - Similar to importing one file with all the preferences, to import a specific RA's preferences, the one line inside the specified CSV file is read and the information is saved into the dictionary.
    - When importing multiple files, each file containing only one RA’s information, the process remains the same. There is only one dictionary that holds all of the RA preferences, so each new file simply adds to the existing dictionary.
- Update Preferences - File:
    - The coordinator imports a file with the updated preferences through the same import button on the user interface.The file contains either a single RA or several RAs. Instead of adding the name and preferences to the dictionary being maintained, the RA’s existing entry in the dictionary will be replaced with the updated preferences.
- Update Individual Preferences - GUI:
    - The user selects which preference for which RA is getting changed, and selects a new preference. On Call Viewer passes this updated information to RA Preferences, and the corresponding field in the dictionary gets changed to the new preference.
- Preferences Settings:
    - The user provides a list of RAs who cannot share the same shift, gives a gold star to one RA, or chooses a method to break a tie in preferences to On Call Viewer. This information is passed to RA Preferences which saves these settings under their own key in the preferences dictionary.

- Undo Preferences:
    - On Call Viewer triggers RA Preferences when the user wants to undo an update made to RA Preferences. This cannot be used for deleting an RA as that is a permanent change to the system.
- Delete RA:
    - On Call Viewer calls RA Preferences when the user wants to permanently delete an RA from the system. This will remove the designated RA from the RA Preferences dictionary.
- Reset Preferences:
    - Removes all of the RA preferences that were inputted into the system by overwriting the existing preferences dictionary with an empty dictionary.

4.2.2. Interface Specification

- RA Preferences interacts with On Call Viewer directly and interacts indirectly with Weekday Scheduler and Weekend Scheduler.
- RA Preferences receives the name of a file from On Call Viewer and reads, parses, and saves the information. It returns either a success or error indicator to On Call Viewer.
- RA Preferences provides a way for On Call Viewer to update a specific field within the dictionary that RA Preferences maintains. It also provides a way to undo the most recent update.
- RA Preferences receives a student ID from On Call Viewer that the user wants to delete and removes the key/value pair from the dictionary.
- RA Preferences provides a way for On Call Viewer to delete all of the RAs in the system.
- RA Preferences indirectly interacts with the scheduler modules because they access, without modifying, the dictionary that RA Preferences maintains. On Call Viewer also accesses this dictionary without modifying it. RA Preferences is the only module that is allowed to change the contents of the dictionary.

Models:

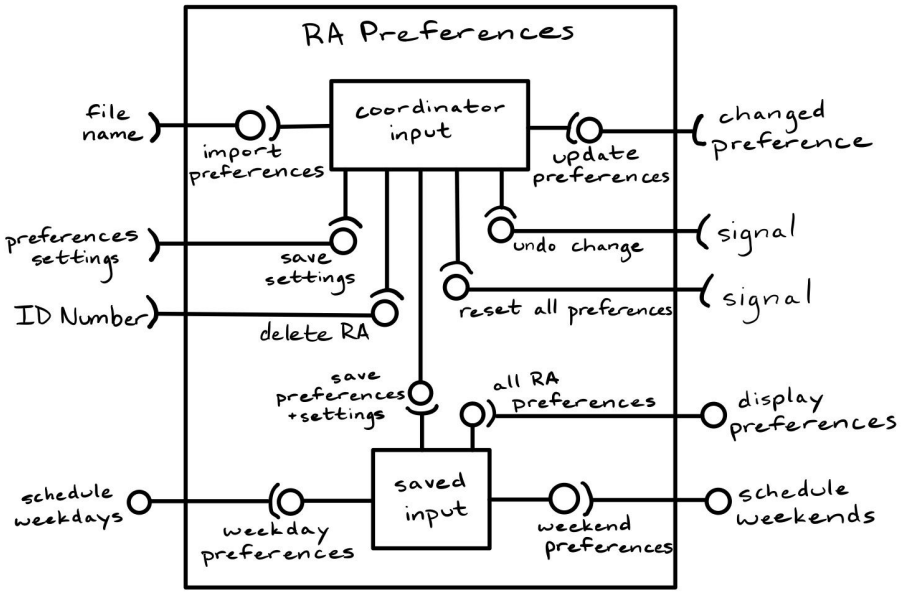

Figure 4. RA Preferences UML Component Diagram.

RA Preferences can receive a file name to import preferences, preferences settings to save in the system, changed preference information to update, and signals to undo changes and reset all preferences. RA Preferences takes input from the user and writes it to a saved dictionary. This saved input can be accessed for weekday preferences to schedule weekdays, weekend preferences to schedule weekends, and all RA preferences to display preferences.

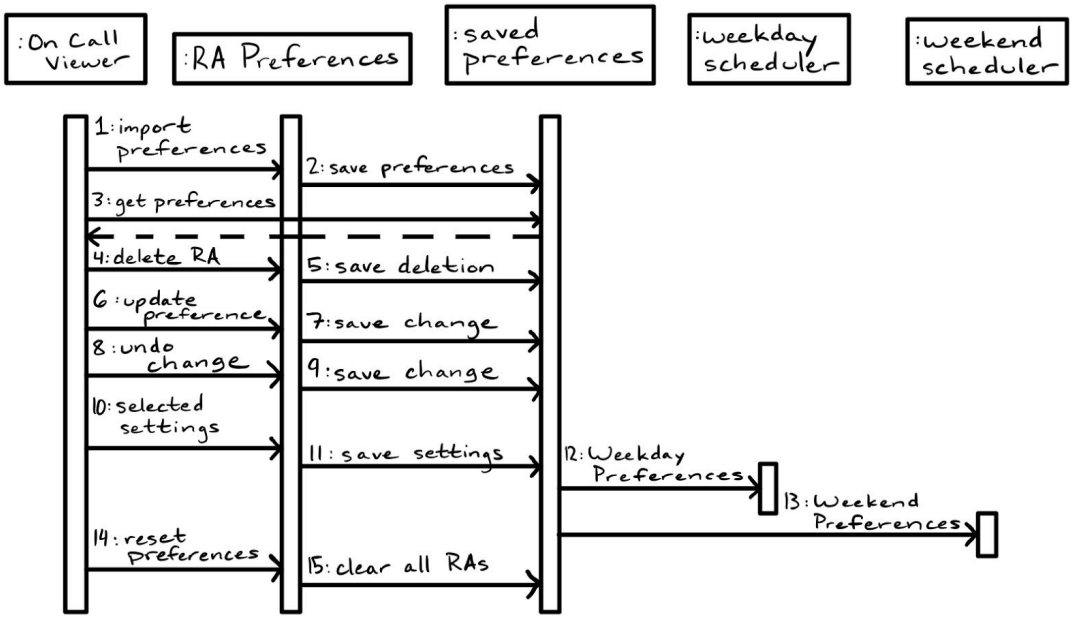

Figure 5 RA Preferences UML Sequence Diagram.

First, the On Call Viewer imports preferences which get processed and saved by RA Preferences. Then On Call Viewer requests the preferences from the saved dictionary. On Call Viewer can then request an RA to be deleted which is saved in the system. Updates to preferences can also be received by RA Preferences and saved. These updates can also be undone by RA Preferences. RA Preferences receives settings to save and then the schedulers access the saved preferences. Finally, RA Preferences can be signaled to reset the saved preferences by clearing all of the saved preference information.

4.2.3. Design Rationale

- RA Preferences is the only module that can alter the preferences saved in the system. This loosens the coupling of the system because there are not several modules all attempting to alter the same information. Since RA Preferences only interacts with the preferences information, the module has strong intramodule cohesion.
- The module provides several different ways for the user to input new and update preferences because various coordinators will prefer different methods. Having all of these methods within the system does not hinder the functionality and improves ease of use.
- This module also provides functionality for the user to undo updates and reset RA preferences to provide users with the option to return to a previous state of RA preferences or to reset the system's RA preferences entirely. This additional feature contributes to ease of use.

4.3. Weekday Scheduler

4.3.1. Primary Function

The Weekday Scheduler module assigns RAs to every weekday shift in a ten week period. Weekday Scheduler utilizes each RA's weekday preferences to assign each RA to a day of the week. The scheduler then decides which weeks each RA is on call and whether they are the primary or secondary RA for that shift.

- Assigning Day of the Week:
    - An algorithm decides which day of the week (Sunday – Thursday) each RA is assigned to by accessing the dictionary containing all RAs and their weekday preferences.
    - The number of RAs assigned to a weekday is as close to even across all five days as possible. For example, if there are sixteen RAs, then four days will have three RAs assigned and the fifth day will have four RAs. Each RA has three weekdays, ranked in order of preference, that they would prefer to be assigned to saved in the dictionary.
    - The algorithm attempts to only assign an RA to a day of the week that is listed in their preferences. The algorithm also attempts to assign as many RAs as possible to their top preference while still being able to fill every weekday with as close to the same number of RAs possible. For those who do not get assigned their top choice, the algorithm attempts to assign them their second choice before resorting to their third choice.
    - If a weekday cannot be filled due to the preferences indicated by each RA, then the algorithm uses the tiebreaking method specified by the preferences settings to handle the tie. A temporary dictionary is created to save which RAs are assigned to which weekdays.
- Assigning Weeks:
    - Using the temporary dictionary that indicates which RAs are assigned to a specific weekday, the RAs randomly get assigned to the weeks they are on call. Checks are made to make sure that the RAs are not a bad pair, or that they have been scheduled more than other RAs assigned to the same day. Primary and secondary shifts are kept as even as possible throughout RAs. This creates another temporary dictionary that logs which two RAs are assigned to each weekday shift.
- Assigning Primary and Secondary RAs:
    - The last part of scheduling the weekday shifts is designating which RA is the primary versus secondary RA during each shift. An RA should have a balance between the number of times they are primary versus secondary RAs.
    - Using the temporary dictionary produced when assigning weeks, the RAs assigned to each shift get marked as primary or secondary. Another field in the dictionary is used to determine the number of times an RA has been primary and secondary.
    - Based on this field, as the algorithm continues through the weeks to assign primary and secondary, the RA who is chosen as primary becomes less random and more based on the number of times in the previous weeks they were marked as either primary or secondary.

- When the algorithm finishes, the dictionary containing which RAs are assigned to each shift, along with their primary and secondary designations, are sent to Shift Assignments to get saved into one schedule.

4.3.2. Interface Specification

Weekday Scheduler interacts with Shift Assignments directly and RA Preferences indirectly.

- Weekday Scheduler passes its temporary dictionary of weekday shift information to Shift Assignments which will save the information into the dictionary containing the entire schedule.
- Weekday Scheduler indirectly interacts with RA Preferences because it will access the dictionary RA Preferences maintains.

Models:

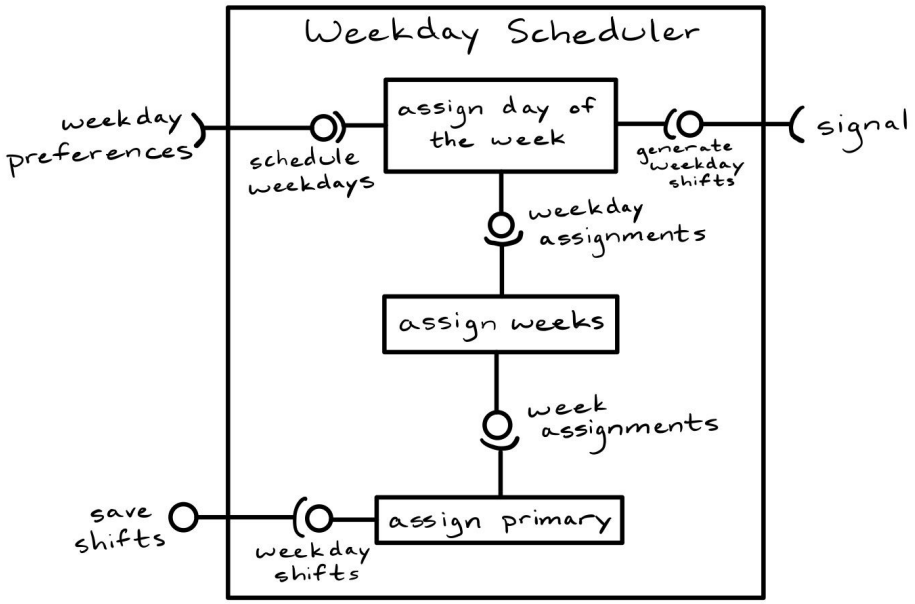

Figure 6. Weekday Scheduler UML Component Diagram.

The Weekday Scheduler receives a signal to generate weekday shifts as well as the weekday preferences to schedule the weekday shifts. Weekday Scheduler sends the assigned weekday shifts to be saved.

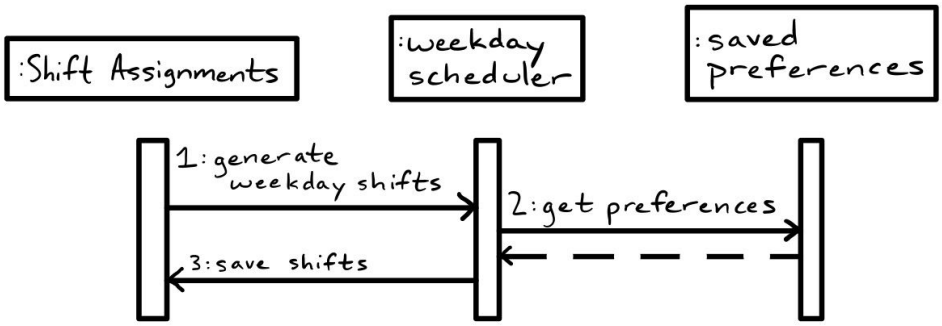

Figure 7.Weekday Scheduler UML Sequence Diagram.
First, Shift Assignments triggers Weekday Scheduler to generate weekday shifts. Then Weekday Scheduler receives preference information from the saved dictionary. Weekday Scheduler runs its algorithm, and then it sends the weekday shift information to Shift Assignments to get saved.

4.3.3. Design Rationale

- Weekday Scheduler does not interact with On Call Viewer in order to maintain loose coupling between modules and strong cohesion within modules. Both schedulers need access to the RA preferences saved in the system. While having a separate module maintain the preferences dictionary means both schedulers need to interact with a separate module, the two schedulers do not need to continually interact with each other to maintain the information- creating looser coupling overall. The cohesion is also strengthened because the schedulers focus on creating the schedules and not maintaining the preferences dictionary.
- Weekday Scheduler is broken down into three parts. All three parts are necessary to schedule the weekdays, so they are included in the same module. They are also distinct from the functions in Weekend Scheduler.
    - Since the system knows the ranked weekday preferences for each RA, the algorithm must start with assigning the day of the week. The algorithm cannot assign primary and secondary markers until each shift has its assigned RAs.
    - This is why the next step is to assign the weeks each RA works, which fills all of the shifts.
    - Finally, the algorithm is able to assign primary and secondary status. Assigning the primary and secondary markers for the weekdays is separate from assigning the same markers for the weekends because each RA should have a balance of primary and secondary on call shifts specifically pertaining to their weekday shifts.

4.4. Weekend Scheduler

4.4.1. Primary Function

The Weekend Scheduler module assigns RAs to every weekend shift in a ten week period. Weekend Scheduler utilizes each RA's preferences for weekends off to assign weekends to each RA. Then the scheduler decides which shift each RA works in their assigned weekends. Finally, the RAs assigned to each weekend shift are designated as primary or secondary RA on call.

- Assigning the First Weekend:
    - The algorithm must account for a balance between the total number of shifts and number of times being primary versus secondary. It copies the RA Preferences dictionary to a temporary dictionary with extra elements to serve as the indexes for the total, primary and secondary shift count.
    - The first weekend is unique as there is no Sunday shift. For each of three shifts the first RA is chosen randomly and if they do not have the week off are assigned as the primary RA on call. The secondary RA for a shift is chosen randomly and rechosen if they have the week off, are together are a user defined bar pair, or if they already have a shift during week one.
- Assigning Weekends 2-10:
    - Weeks 2 through 10 each have four shifts. For each shift the first RA is randomly chosen as the Primary and rechosen if they have more primary than secondary shifts or more shifts than average or have the week off. The second RA is randomly chosen as the secondary on call and rechosen if together they are a user defined bad pair, the RA has more secondary than primary shifts or more shifts than average.
    - When the algorithm finishes the list of two lists containing the entire ten week schedule of which RAs are assigned to each shift, along with their primary and secondary designations, are sent to Shift Assignments to get saved in the overall schedule.

4.4.2. Interface Specification

Weekend Scheduler interacts with Shift Assignments directly and RA Preferences indirectly.

- Weekend Scheduler passes its temporary 3D list that contains weekend shift information to Shift Assignments which saves the information into the dictionary containing the entire schedule.
- Weekend Scheduler indirectly interacts with RA Preferences because it simply accesses the dictionary that RA Preferences maintains. It does not call any functions provided by RA Preferences.

Models:

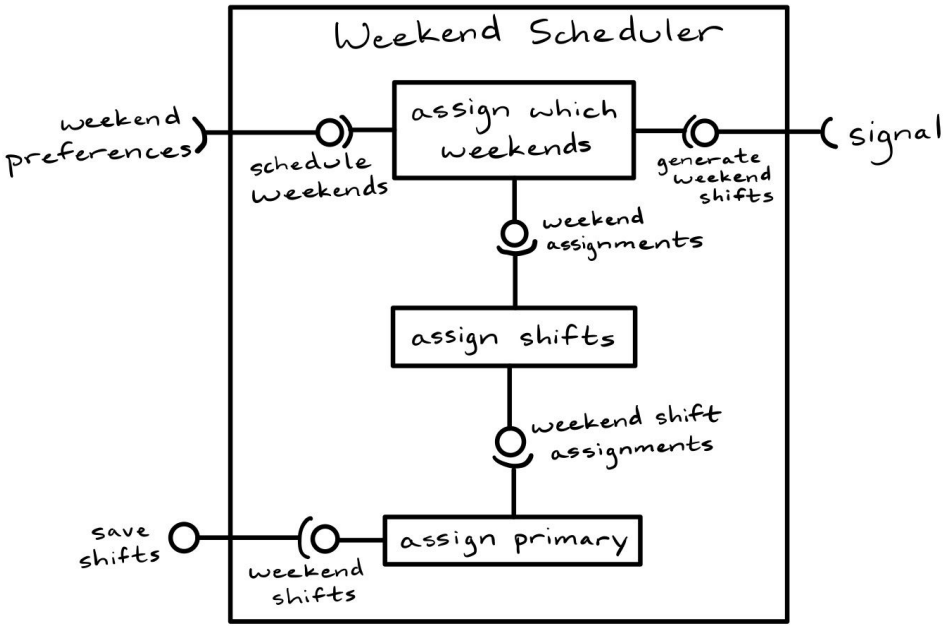

Figure 8. Weekend Scheduler UML Component Diagram.
The Weekend Scheduler receives a signal to generate weekend shifts as well as the weekend preferences to schedule the weekend shifts. Weekend Scheduler sends the assigned weekend shifts to be saved.

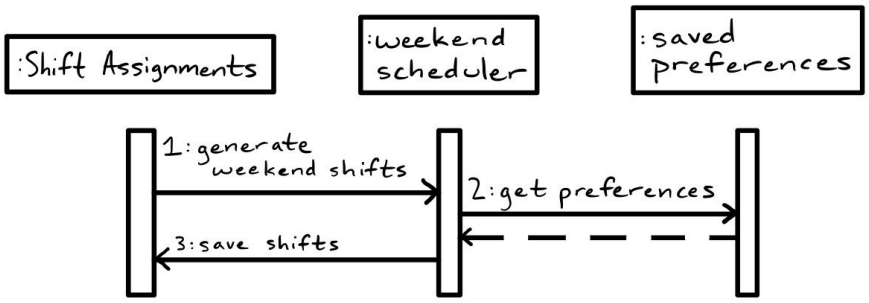

Figure 9. Weekend Scheduler UML Sequence Diagram.
First, Shift Assignments triggers Weekend Scheduler to generate weekend shifts. Then Weekend Scheduler receives preference information from the saved dictionary. Weekend Scheduler runs its algorithm, and then it sends the weekend shift information to Shift Assignments to get saved.

4.4.3. Design Rationale

- Weekend Scheduler does not interact with On Call Viewer to maintain loose coupling between modules and strong cohesion within modules.
- Both scheduler modules need access to the RA preferences saved in the system. While having a separate module maintain the preferences dictionary means both schedulers need to interact with a separate module, the two schedulers do not need to continually interact with each other to maintain the information- creating looser coupling overall. The

cohesion is also strengthened because the schedulers focus on creating the schedules and not maintaining the preferences dictionary.

- Weekend Scheduler is broken down into three parts. All three parts are necessary to schedule the weekends, so they are included in the same module. They are also distinct from the functions in Weekday Scheduler.
    - The algorithm must guarantee each RA gets all three of their weekend off requests. This means the algorithm needs to start by assigning weekends to each RA. It cannot start with assigning shifts within weekends because of this.
    - The next step is to assign the shifts with each weekend. Starting with week one. The algorithm assigns primary and secondary RAs until each shift has its assigned RAs.
    - Finally, the algorithm is able to balance the number of primary and secondary shifts as well as the average number of total shifts each RA has. Assigning the primary and secondary markers for the weekends is separate from assigning the same markers for the weekdays because each RA should have a balance of primary and secondary on call shifts specifically pertaining to their weekend shifts.

4.5. Shift Assignments

4.5.1. Primary Function

The Shift Assignments module saves the weekday and weekend shift information into one schedule. The module maintains the on call schedule and allows the user to export it to a CSV file. Functionality to update the schedule, undo updates to the schedule, and reset the schedule are also provided.

- Generate Schedule:
    - The user can request the system to generate a new schedule through On Call Viewer.
    - Upon this request, Weekday Scheduler and Weekend Scheduler will run. These modules both produce temporary dictionaries that will be passed to Shift Assignments.
    - The information contained in these separate dictionaries is then inserted into a single dictionary that is saved between sessions. To save the dictionary between sessions, it will be written to a Python file.
- Update Schedule:
    - The user can select the name assigned to a shift and enter a different name through On Call Viewer.
    - The information on which shift needs to be updated and what it needs to be updated to is passed to Shift Assignments.
    - The dictionary with the information on the overall schedule is then updated appropriately.
- Undo Schedule:
    - On Call Viewer triggers Shift Assignments when the user wants to undo an update made to the schedule.

- Export Schedule:
    - Using the specified name and location of the CSV file, the information contained in the dictionary of all of the shift assignments is written to the file. The file will have a header row that indicates the day of the week.
    - Then, each line written to the file will be the full names of RAs assigned to be on call that week—matching the names with the correct day of the week columns.
    - Each week gets two lines because the first line is the names of the primary on call RA and the second line is for the secondary on call RAs. The beginning of each line has a field that specifies the week and primary or secondary. This formatting lets the user open the CSV in another application like Excel and see a schedule that looks similar to a calendar format.
- Export Summary:
    - On Call Viewer passes the name of a text file for the summary of each RA's shifts to be exported to.
    - The schedule is looped through to count how many shifts of each type every RA is assigned.
    - The schedule is also looked at to determine which day of the week and which weekend shift every RA is assigned to.
- Reset Schedule:
    - Removes the generated schedule by overwriting the schedule with an empty schedule.

4.5.2. Interface Specification

- Shift Assignments interacts with both Weekday Scheduler and Weekend Scheduler in similar ways.
    - Both of the schedulers will send lists of shift information to Shift Assignments which will insert the information into a dictionary of the entire schedule. Shift Assignments is also the module that triggers the schedulers to run.
- Shift Assignments also interacts with On Call Viewer.
    - On Call Viewer can request Shift Assignments to generate a new schedule.
    - Shift Assignments can receive the name of a file from On Call Viewer to export the full schedule to.
    - Shift Assignments can receive the name of a file from On Call Viewer to export a summary to.
    - Shift Assignments can be triggered by On Call Viewer to undo an update made to a shift in the schedule or to delete the entire schedule.
    - Shift Assignments indirectly interacts with On Call Viewer when On Call Viewer access the dictionary containing the schedule.

Models:

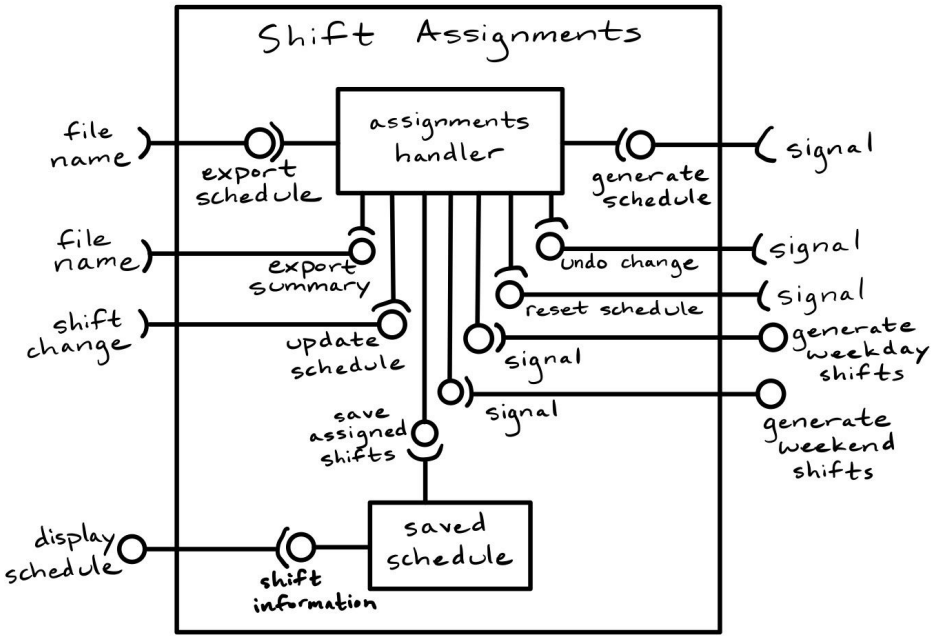

Figure 10. Shift Assignments UML Component Diagram.

Shift Assignments receives shift change information to update the schedule, file names to export schedules and summaries, and signals to generate schedules, undo changes, and reset schedules. Shift Assignments sends signals to generate weekday shifts and weekend shifts. Shift Assignments saves the shift information into a dictionary which can be accessed to display the schedule.

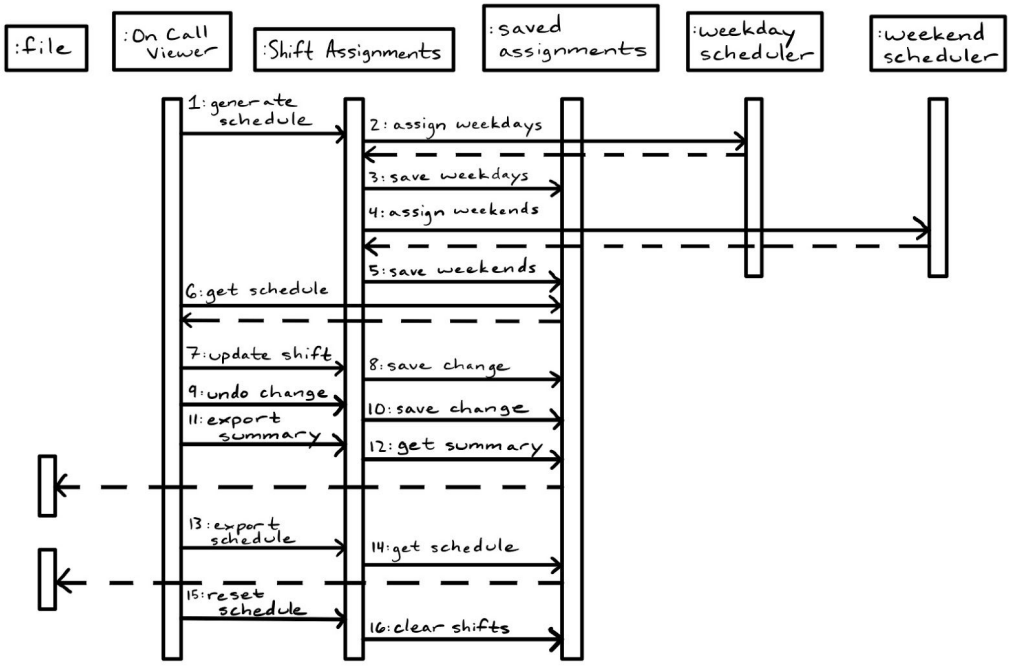

Figure 11. Shift Assignments UML Sequence Diagram.
First, On Call Viewer triggers Shift Assignments to generate a schedule. Then Shift Assignments requests weekday assignments and saves them into the schedule. Next, Shift Assignments requests weekend assignments and saves them into the schedule. When On Call Viewer wants to display the schedule, it gets it from the saved dictionary. On Call Viewer can send Shift Assignments information to update a shift which gets saved as well as trigger Shift Assignments to undo that change. On Call Viewer can request Shift Assignments exports the summary and schedule which get written to files. Finally, On Call Viewer can trigger Shift Assignments to delete the saved schedule.

4.5.3. Design Rationale

- Shift Assignments is the only module that can alter the schedule saved in the system. This loosens the coupling of the system because there are not several modules trying to alter the same information. Shift Assignments only handles schedule information, so the module has strong intramodule cohesion.
- This is the module that triggers the scheduler modules to run because it will take the temporary lists returned by the schedulers to create the saved dictionary of the entire schedule.
- This module provides a way for the schedule to be exported because the coordinator needs to be able to send the schedule out to the RAs. Shift Assignments handles this as it is the module in charge of handling schedule information.
- This module handles creating the summary because, once again, it is the module that handles functionality related to the shift assignments.
- Since Shift Assignments is the only module that can alter the schedule saved in the system, this is the module that handles updating, undoing, and resetting shift assignments. Updating is important because sometimes the coordinator needs to swap shifts around. This means undoing is also important because the coordinator might make a mistake when changing shifts. Resetting is helpful if the coordinator simply wants to clear the system of all saved information.

4.6. Alternative Designs

- On Call Viewer provides a way for the coordinator to enter a new RA by typing the information in the GUI instead of importing a file. This seemed error prone because the coordinator could miss type the information when trying to copy it from another file. Enforcing the file upload for a new RA reduced the chance for errors.
- On Call Viewer allows the coordinator to update RA preferences by importing files and not by typing in the GUI. It seemed more user friendly to allow the preferences to be updated through the GUI. This is not as error prone as entering the information for an entirely new RA.
- To settle a tiebreaker, the coordinator gets a prompt while the scheduler is running and selects which RA is assigned instead of choosing a selection method before running the schedulers. This would cause tighter coupling because the schedulers would have to interact with On Call Viewer.
- The coordinator can only view the schedule by exporting the CSV file and opening it in another program such as Excel. This also meant any updates to the schedule were not saved in *On Call*. To improve user friendliness, the coordinator can now view and update the schedule in *On Call*.

4.6.1. On Call Viewer Original Models

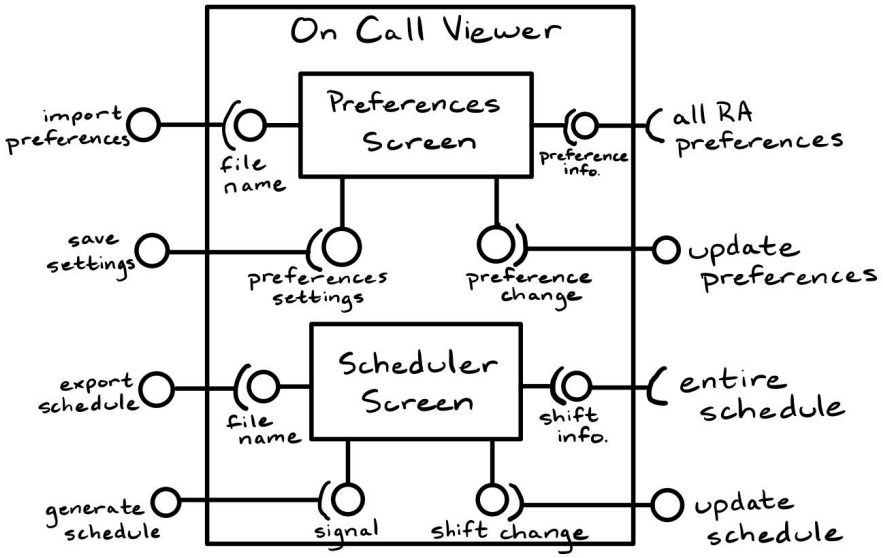

Figure 12. On Call Viewer UML Component Diagram iteration one.
This diagram does not have the interactions pertaining to deleting one or all RAs nor undoing an update. It also misplaces which portion of On Call Viewer handles saving the preferences settings. Finally, this diagram does not include exporting a shift summary, undoing a schedule change, and resetting the schedule.

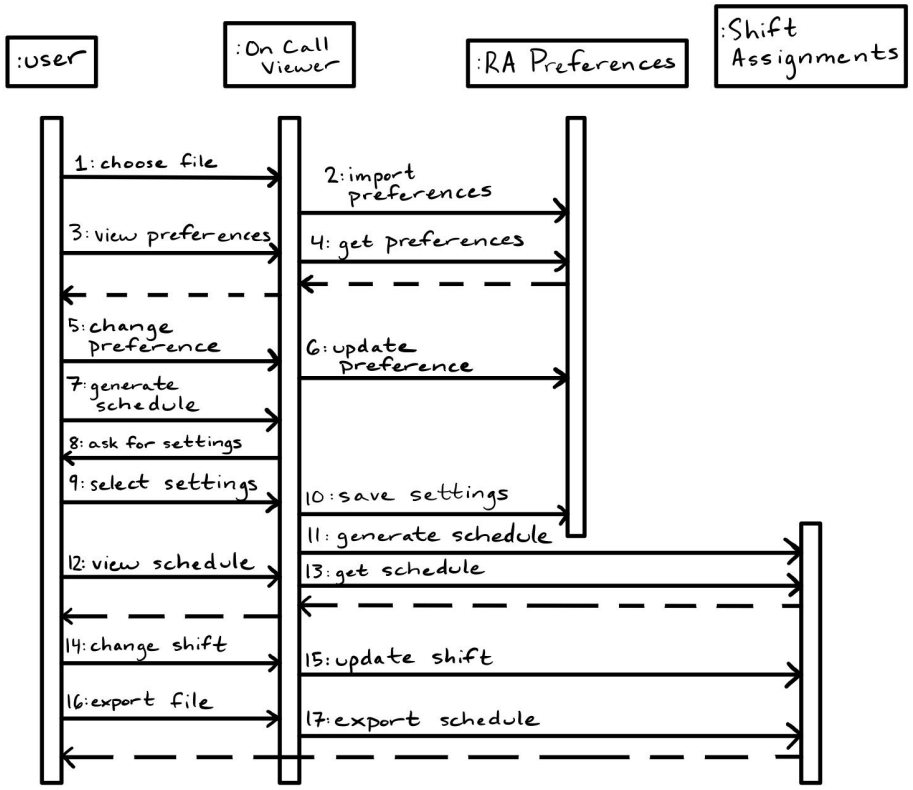

Figure 13. On Call Viewer UML Sequence Diagram iteration one.
This diagram excludes deleting a single RA, clicking undo on preferences, clicking undo on shifts, exporting a summary, resetting preferences, and resetting shifts.

4.6.2. RA Preferences Original Models

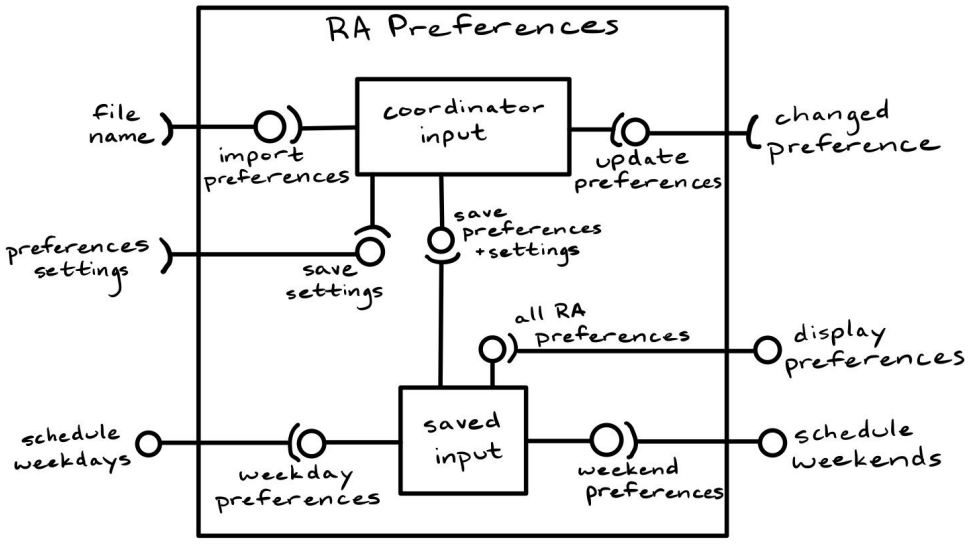

Figure 14. RA Preferences UML Component Diagram iteration one.
This diagram excludes receiving an ID number to delete an RA, receiving a signal to undo a change, and receiving a signal to reset all preferences.

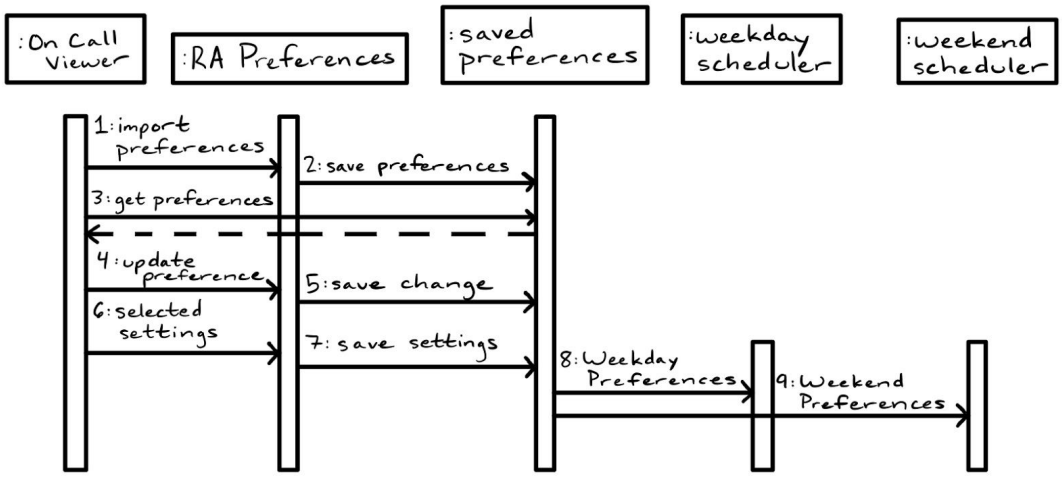

Figure 15. RA Preferences UML Sequence Diagram iteration one.
This diagram excludes deleting a single RA, undoing an update, and deleting all RAs.

4.6.3. Shift Assignments Original Models

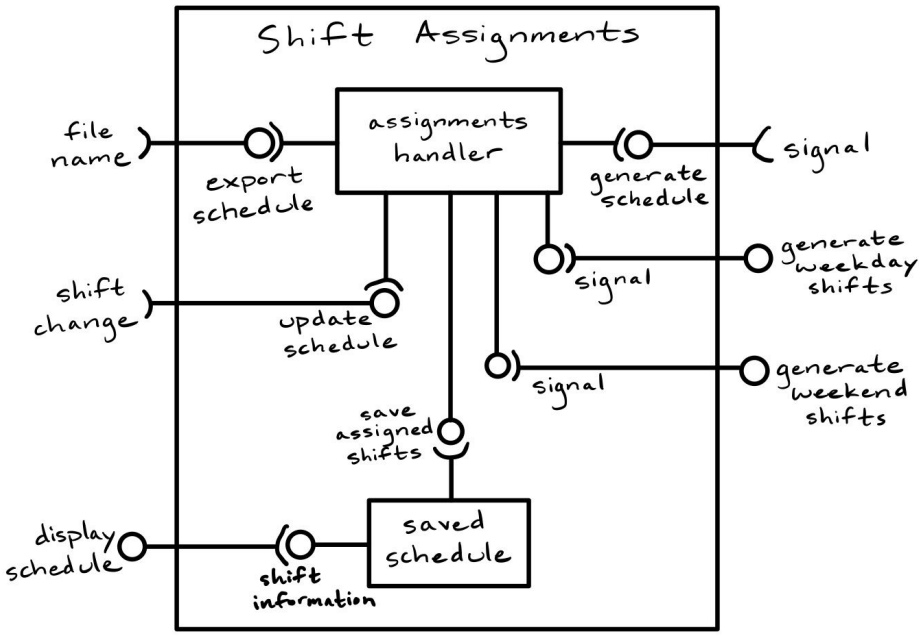

Figure 16. Shift Assignments UML Component Diagram iteration one.
This diagram excludes receiving a file name to export a summary, a signal to undo a change, and a signal to reset the schedule.

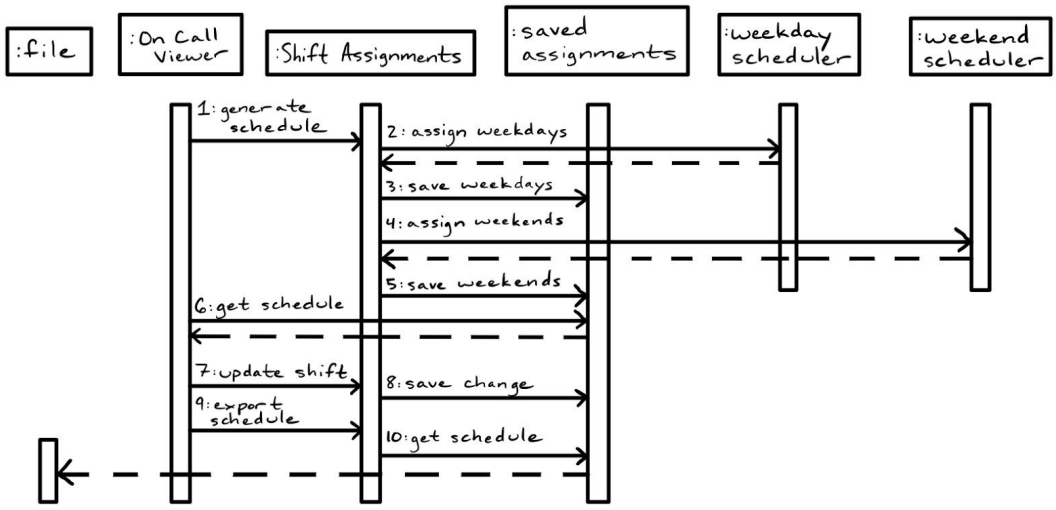

Figure 17. Shift Assignments UML Sequence Diagram iteration one.
This diagram excludes undoing a change, exporting a shift summary, and resetting the schedule.

5. Dynamic Models of Operational Scenarios

5.1. Use Case A: Resident Assistants Input Shift Preferences

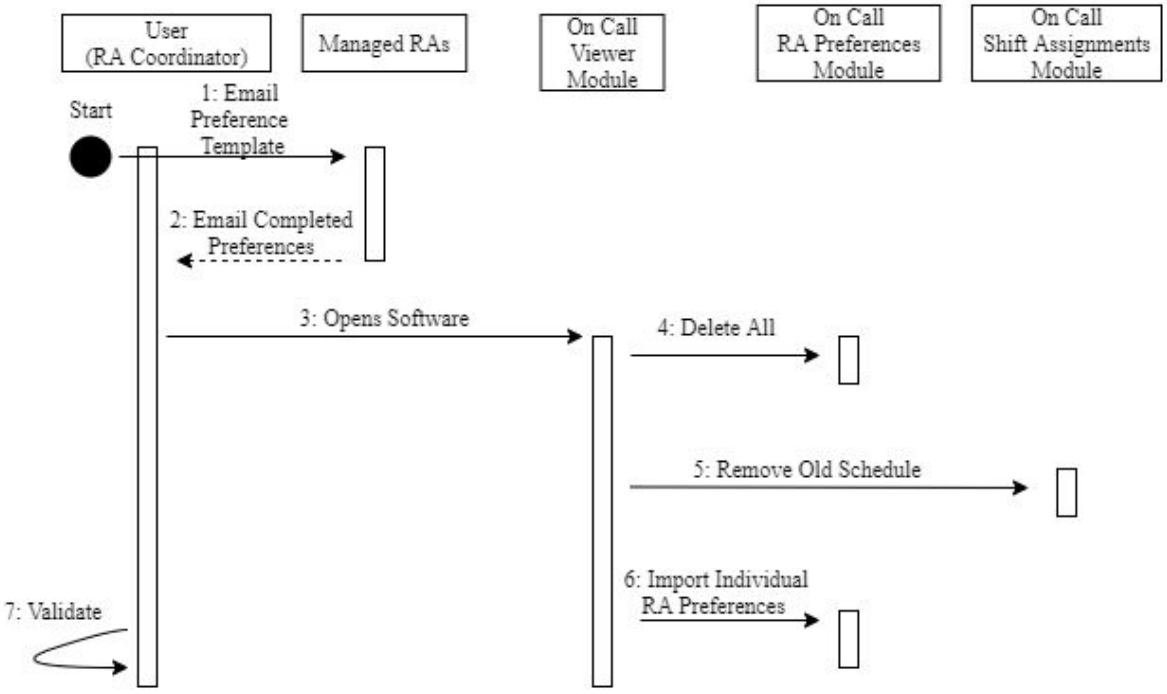

Figure 18. UML Sequence Diagram for Resident Assistants Input Shift Preferences. The user sends an email containing the RA preference template for RAs to complete. Once the RAs email their preferences back to the user, they open *On Call* to be greeted by the On Call Viewer module. The user imports the preferences to the software in the form of a CSV file.

5.2. Use Case B: Running the Program to Create the Term Schedule

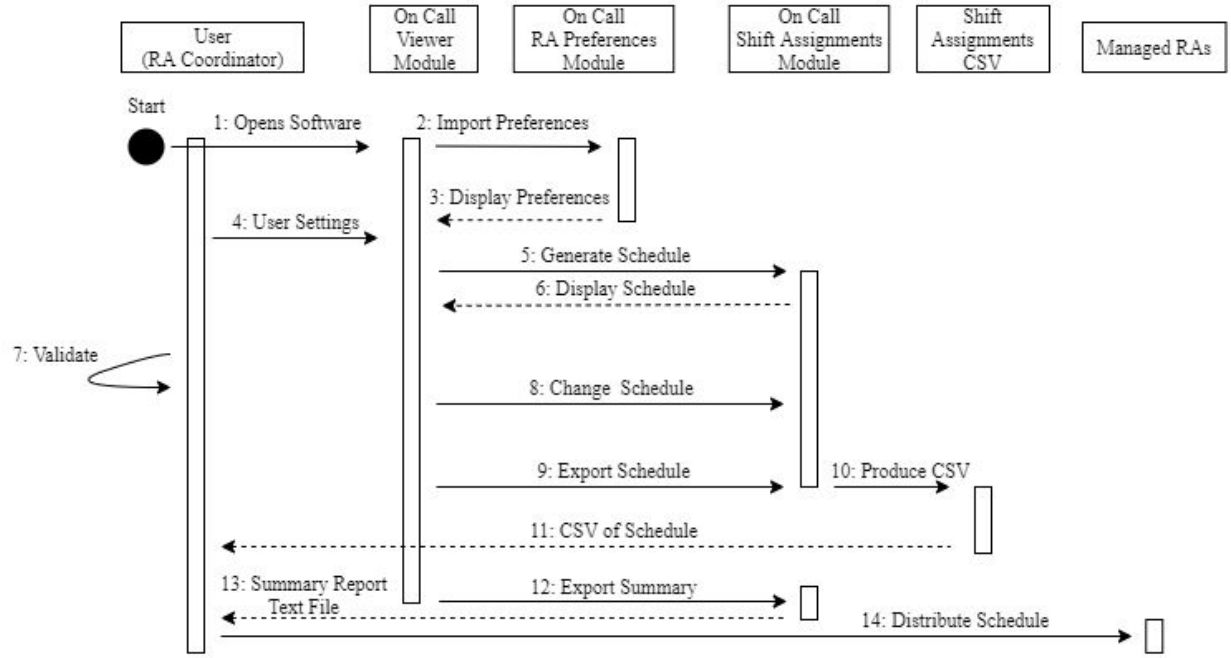

Figure 19. UML Sequence Diagram for Running the Program to Create the Term Schedule. The user opens the software and imports the RA preferences. *On Call* displays the preferences that were imported. The user then opens user settings and clicks on the button that generates and returns the shift assignments schedule to be displayed on the *On Call* interface. Users can validate and make changes to the schedule. They can press the export button which produces a CSV file of the shift assignments that is saved under the user's desired file name. This schedule can then be distributed to the RAs.

5.3. Use Case C: Updating RA Preferences and Shift Assignments

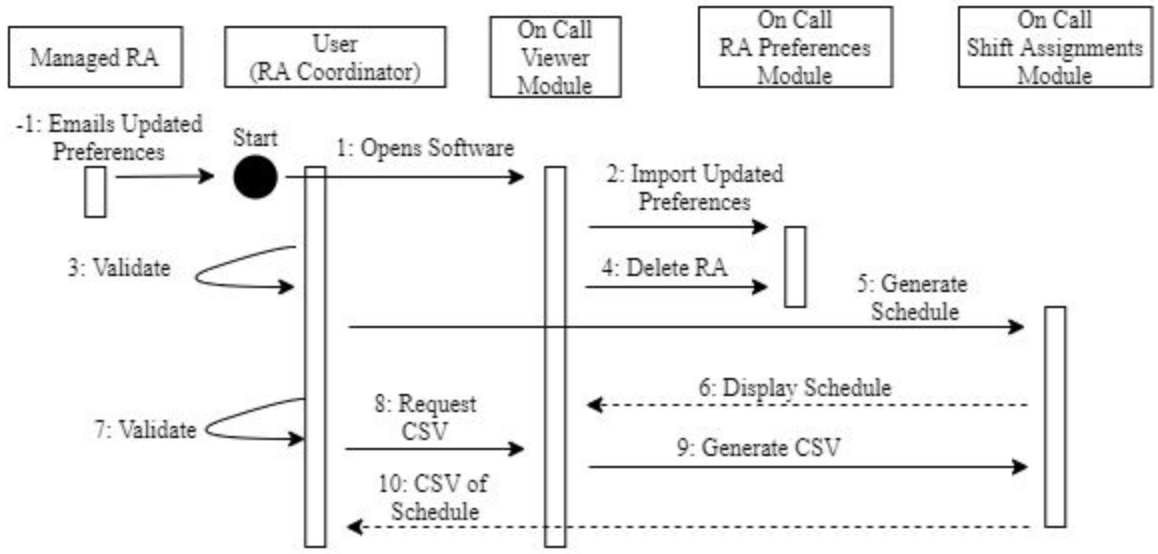

Figure 20. UML Sequence Diagram for Updating RA Preferences. If the user receives an email from a RA who wants to update their preferences, the user opens *On Call* and imports the update RA preferences. Then, the user generates an updated schedule and can export the updated CSV file of the shift assignments schedule.

5.4. Use Case D: Undo RA Preferences and Shift Assignments

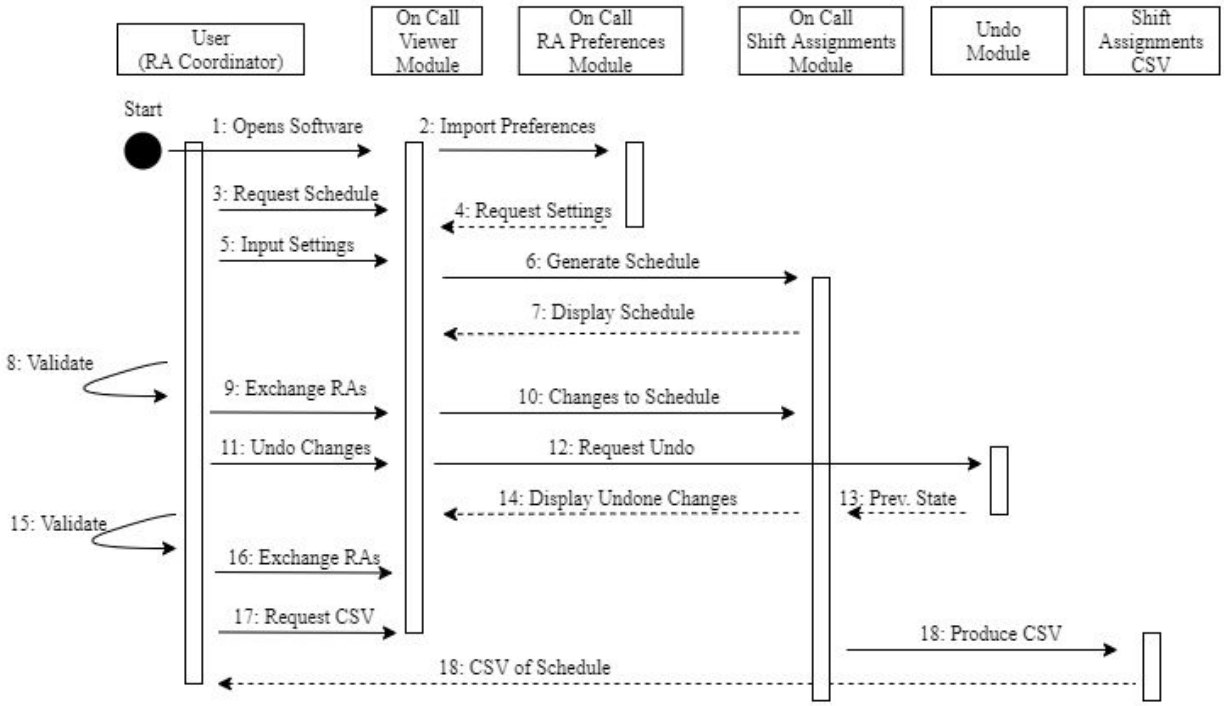

Figure 21. UML Sequence Diagram for using the Undo function.The user opens *On Call* and imports RA preferences and requests a schedule be generated. The program will prompt for user settings to be input before a schedule can be generated. After viewing the schedule the User can make changes to the shift assignments in the GUI. Such changes can be undone with the Undo button to revert to previous states. Upon satisfactory review the user can export the CSV of the schedule for distribution.

5.5. Original Use Case Diagrams

After the initial submission of requirements, some changes were made. The following diagrams were created for the original use cases.

5.5.1. Original Use Case A Diagram

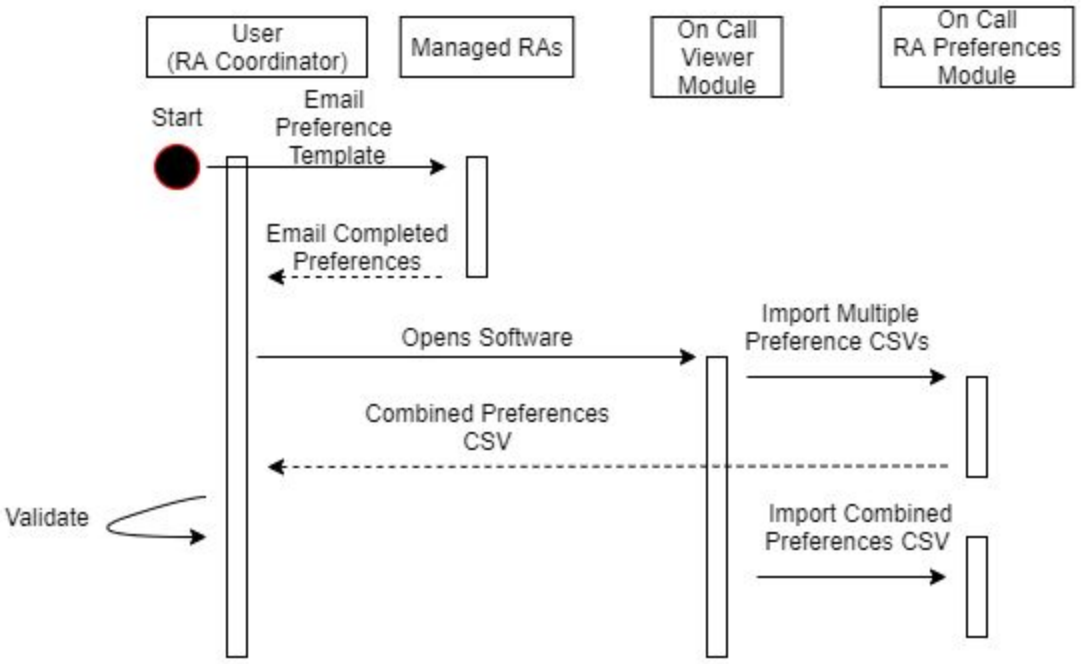

Figure 22. The oldest version of the UML Sequence Diagram from 5.1.1. There were changes to labeling, adjustments of timings, fonts and the retirement of extras.

5.5.2. Original Use Case B Diagram

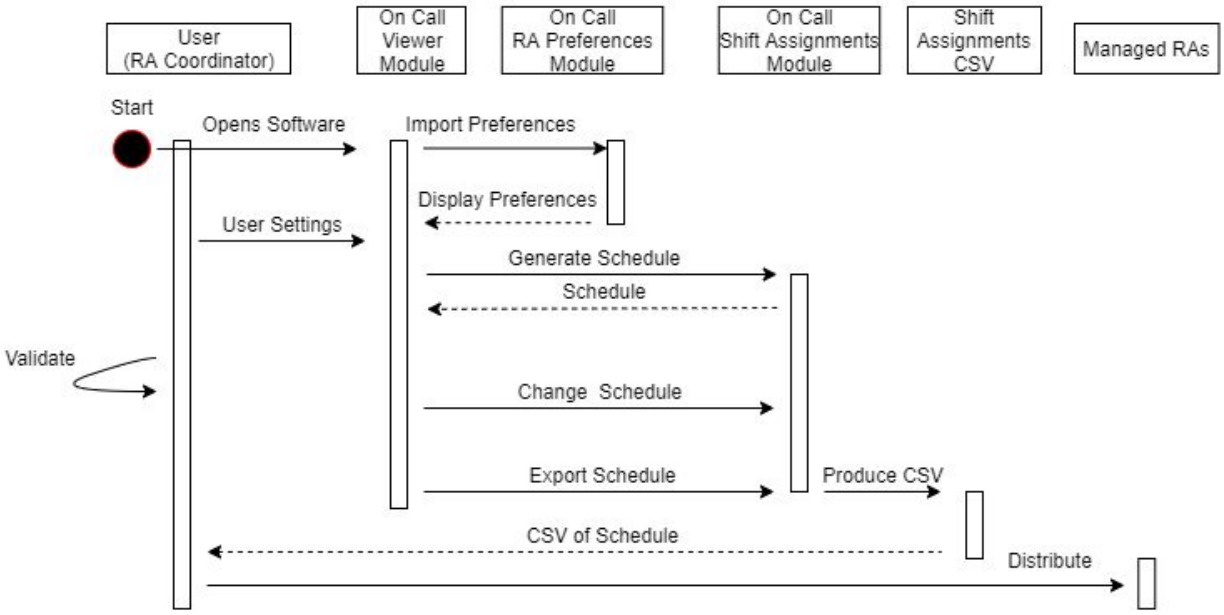

Figure 23. The oldest version of the UML Sequence Diagram from 5.2.1. There were changes to labeling, adjustments of timings and fonts. In particular the user must request the CSV be exported before a CSV of the schedule exists.

5.5.3. Original Use Case C Diagram

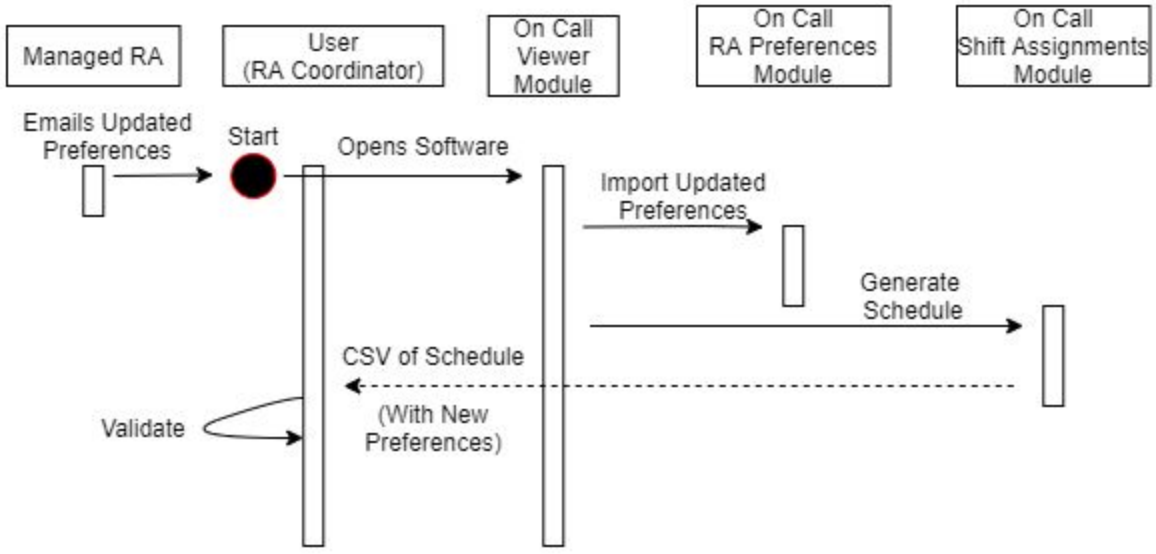

Figure 24. The oldest version of the UML Sequence Diagram from 5.3.1. There were changes to labeling, adjustments of timings and fonts.. In particular the user must request the CSV be exported before a CSV of the schedule exists.

6. Diagram Keys

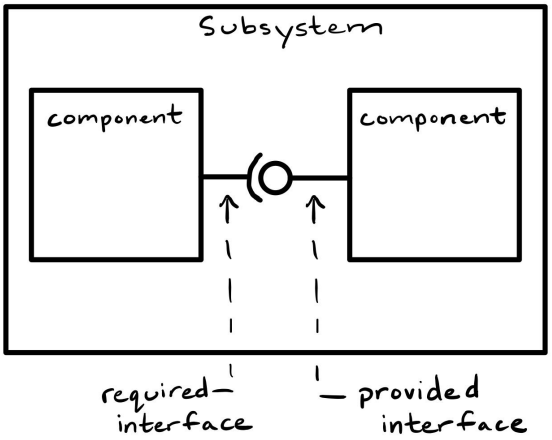

Figure 25. Section 4 UML Component Diagram Key.
The outer box represents the subsystem and the inner boxes represent the components of the subsystem. The line with the half circle represents the required interface, and the other line with the full circle represents the provided interface.

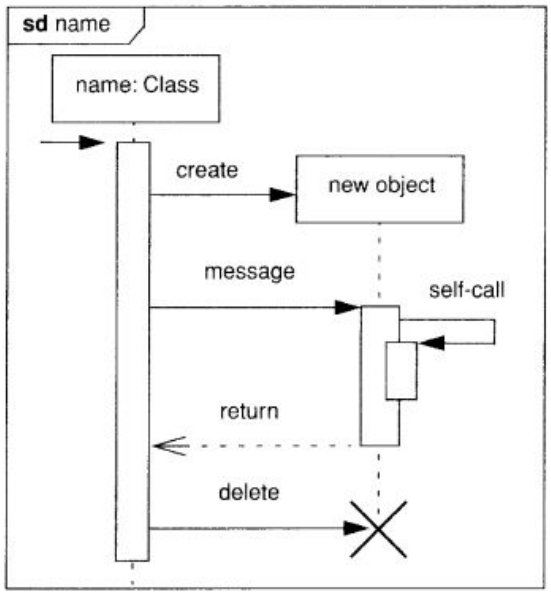

Figure 26. Section 5 UML Sequence Diagram Key.
(Hornof, Anthony and Martin Fowler. 2020)

7. References

draw.io - Online Diagramming. (2020). Accessed at https://about.draw.io/.

Hornof, Anthony and Martin Fowler. (2019). CIS 422 Lecture Notes. Downloaded from https://classes.cs.uoregon.edu/20W/cis422/Handouts/Lecture_Notes.pdf in 2020.

Hornof, Anthony. (2019). CIS 422 Software Design Specification Template. Downloaded from https://classes.cs.uoregon.edu/20W/cis422/Handouts/Template-SDS.pdf in 2020.

Visual Paradigm. (2020). What is Component Diagram?. Accessed at https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/ in 2020.

Visual Paradigm. (2020). What is Sequence Diagram?. Accessed at https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/ in 2020.

8. Acknowledgements

Section 5 uses templates and software provided by draw.io.

Figure 15 is copied from CIS 422 class lecture notes.

This document uses a template provided by Anthony Hornof in CIS 422 at the University of Oregon.

To create the section 4 models, the site Visual Paradigm was referenced for formatting purposes.