**Technical Manual**

1. **ATDS**
   1. **Hashtable**

The Hashtable is created in the createHashtable() method in DiningHallSelector.java and is utilized in the createMenu() method in the DiningHall.java class. The Hashtable holds our data from a .tsv file, the data is made up of a Dish’s name, and a score. The Hashtable contains the key-value pair of a String that is a menu item’s title, and a Dish object representing that menu item with its name and score. If the Hashtable contains the dish’s title on that day’s menu, then the Dish object value is retrieved from the Hashtable and added to that dining hall’s daily menu. We chose to use a Hashtable because of the O(1) check to see if a key is in the Hashtable, and fast retrieval if the object is present in the Hashtable.

* 1. **Priority Queue**

We had difficulty implementing the Java implementation of a Priority Queue, so we elected to use the Java Foundations implementation. Therefore, our Priority Queue uses a Max Heap, which turned out to be ideal for our program. We used two Priority Queues. The first Priority Queue contains a DiningHall object’s menu in the createMenu() method DiningHall.java. We chose a Priority Queue for the menu primarily because we wanted an easy way to remove the highest rated element. Also, we wanted a structure that gave us flexibility in how many elements we added to it. The second Priority Queue is a queue of DiningHall objects in the DiningHallSelector.java class, and it’s used in the getTopTwoDiningHalls() method. The Priority Queue makes it easy for us to retrieve the two highest rated dining halls.

* 1. **Linked List**

We used a Linked List to hold all of the dishes on the Wellesley Fresh menu in the readWellesleyFresh() method in the DiningHallSelector.java class. We chose a Linked List because of its flexible size, the ease, as well as the ease of adding and removing elements.

1. **Important classes (Back-end classes)**
   1. **Dish.java (implements Comparable<Dish>)**
      1. Each Dish object represents a menu item or “dish” on the Wellesley Fresh menu. The Dish class implements the Comparable interface. Every Dish object contains a name and a score. A Dish is greater than another Dish if it has a higher score.
      2. The Dish class contains getters and setters, a compareTo() method, a toString() method, and a method to shortenName() method that just returns the first forty-five characters of the name.
   2. **DiningHall.java (implements Comparable<DiningHall>)**
      1. Each DiningHall objects represents a DiningHall at Wellesley College. All DiningHall objects contain a name for the DiningHall object, a menu represented by PriorityQueue<Dish> for a given meal, and the average score of the items on the menu for that given meal.
      2. Important Methods:
         1. private void addToMenu(Dish d)
         2. private PriorityQueue<Dish> copyMenu()
         3. public void createMenu(String todaysMenuFile, String dataFile, String mealName)
         4. public double calcScore()
         5. public Dish getTop()
         6. public int compareTo (DiningHall d)
         7. public void initializeDiningHall(String todaysMenuFile, String dataFile, String mealName)
   3. **DiningHallSelector.java**
      1. Each time the program is run, a DiningHallSelector object is created representing the process of choosing the best dining hall at a given meal. In the constructor five dining halls are created as representations the five dining halls at Wellesley. This class contains a readWellesleyFresh() method that reads a weekly menu from a txt file as well as a createHashtable() method that reads data from a tsv file that holds Wellesley Fresh common dishes and their student reviews. This class also uses java Calendar API to return an int representing the days of the week at a given date. This class also creates helper methods that neaten text and help retrieve the day of the week.
      2. Important Methods:
         1. public void initializeAll(String mealName) // initialize all the dinning halls for either lunch/ dinner
         2. public static LinkedList<String> readWellesleyFresh (String inFileName, String meal)
         3. public static Hashtable<String,Dish> createHashtable(String inFileName)
         4. public DiningHall[] getTopTwoDiningHalls()
2. **GUI classes**
   1. **SelectorGUI.java** 
      1. The driver class that establishes JFrame with 3 Tabbed Panels: InstructionsPanel, TodaysChoice, JustForYouPanel.
   2. **InstructionPanel.java**
      1. InstructionsPanel is the first panel that users view. It contains the general instructions of the program and lovely pictures of all the dining halls.
   3. **TodaysChoice.java**
      1. TodaysChoice contains both lunch panel and dinner panel for user to click on the lunch/ dinner button to see the corresponding results generated through the program. The results are calculated based on comparing today's menu to the hashtable we have in the database. The program will generate two best dining halls for each meal.
   4. **JustForYouPanel.java**
      1. JustForYouPanel contains both lunch panel and dinner panel. In each panel, it contains checkboxes of the main dish from every dining hall for users to check the ones that they like. For the dish that are checked by the user, the average score of the dining hall that serves the dish increase by 1 in calculating the top two dining halls. The program will generate two best dining halls for each meal.