Elizabeth Hau and Emily Cetlin
Phase 1: Specifications
Deliverables: User's Manual and Technical Report [10 points]
Due: Thursday, May 1, 2014

1. The **User's Manual** of what you expect will be the overall behavior of your program. For the GUI (graphical user interface), draw a picture of what you expect the interface to look like. Explain how the user will be able to interact with your program.

For our final project, we hope to design a program that can complete the Housing Lottery process by sorting Wellesley students into their dorms once they are given their lottery number. The lottery works as follows: students are given their lottery numbers and options for what type of housing they can choose from. Rising sophomores are allowed to be in doubles and rising juniors and seniors can be in singles or doubles. Once students are given a lottery number, they can decide who they would like to room and/or block with. From there, students take the average of everyone in their block, list their dorm preferences (the dorm preferences for students in the same block must be the same), staple the forms together, and turn them in. The average becomes the group's new lottery number. The housing department then places the blocks of students into dorms based on their lottery number. Blocks with the best lottery numbers are placed first. If there are enough rooms available in the block's top choice dorm, they will be placed there. Otherwise, the housing department will then look to their second choice, third choice, etc. If a block does not get into any of the rooms they have placed down, they will become unhoused and their rooming assignment will be determined at a later date.

The intended user for this project is someone in the housing office. Our program assumes that the housing office either has all of the forms that the students have submitted, or has a file with all of the information collected from the students across campus. The user can either input the file or can individually add students and blocks. The user can also input a file and then add students. Our GUI has three tabs: an Instructions tab, an Add Student tab, and a Results tab. On the Instructions tab, the user can learn about how the program works and enter a file. On the Add Student tab, the user can input a student and their block. On the Results tab, the user can calculate the results of the lottery and see where students are placed. See the attached document for a drawn draft of the GUI design.

2. A **Technical Report** with description of:

2a. The **ADTs** that will be used and what information they will store. Include a brief justification for each of your choices.

The two ADTs we will be using are a Priority Queue and a Linked List. We will be using the Priority Queue as part of the Lottery class. The Priority Queue will take in all of the blocks and pop out the blocks in order when placing them into dorms. In our Block class, we will have one Linked List of type Student and another of type String. The list of type Student contains all the students in the block and each student's respective information. The list of type String would contain the names of the dorms. We decided to use LinkedList because we don't know exactly how many students would be in each block

(although there is a minimum of 1 and a maximum of 6 students per block). Moreover, we don't know how many dorm names will be in their preferences, so using a LinkedList would allow us to add to the list as needed and would avoid allocating unnecessary space.

2b. A list of the important **classes** that you expect to define for your project with a brief description of the purpose of each class. Some of these classes should capture the basic objects that exist in the problem. There may also be classes that embody the graphical user interface, or the main() method. This list should include the classes that implement the ADTs that you plan to use. Note that as you proceed with your program development, you may discover other classes that would be useful to define for your application.

Our program would contain:
- a Dorm class. The Dorm class creates a dorm with a certain number of senior singles, junior singles, sophomore doubles, and first year doubles.
- a Quad class. The Quad class creates the collection of dorms on the Quad. It uses the Dorm class to create each dorm.
- a Student class. The Student class creates each student, containing student name, and lottery number.
- a Blocking class. The Blocking class contains a group of students/student to place into the lottery. It uses the Student class. The class contains two Linked Lists: one of Students and the other of dorm preferences.
- a Lottery class. The Lottery class takes a group of blocks and assigns them to dorms in the Quad using a Priority Queue. This class uses Blocking class and Quad class.
- a LotteryGUI. The LotteryGUI creates an interface that allows a user to easily use the Lottery class to calculate the dorm assignments for a group of students.

2c. A list of some of the main **actions** that you expect to be embodied in methods in your new class definitions (you do not need to include the basic operations defined for the ADT classes that you plan to use). As you proceed with your program development, you will probably discover additional useful methods to define for various classes.

- Dorm
    - getEmptySingles()
    - getEmptyDoubles()
    - addStudent()
- Quad
    - addDorm()
- Student
    - getLotteryNumber()
- Blocking
    - addStudent()
    - calcAvg()
    - preferences()
- Lottery
    - addBlock()
    - calcResult()

Wellesley Housing Lottery

instructions | Add student | Results

Instructions

Welcome to the Wellesley College Housing Lottery Calculator. To calculate the results for next year's housing assignments, please enter the name of a file below or manually input students' information in the "Add student" tab.
You may also use the Add Student tab to add a student to a file. Then go to the "Results" tab to see the results.

Text Box For File | Add File

Add Student

Please enter the following information to add a student

Name: _____ Lottery #: _____

Current year: o First year o Sophomore o Junior o Senior

Blocking:

	Students	Numbers
1:		
2.		
3.		
4.		
5.		

Add student

Group Average: ...

Results

Calculate!