5CCS2SEG Agile Project

Team Jay

Introduction

For the duration of this project we have, of course, followed an ‘Agile’ approach in development. What this entails is a ‘write code now, refactor code later’ mentality where development of the application is more of a ‘sprint’. There is less planning as progress is made than in more traditional development techniques.

In spite of this pace, however, from the moment we first started development of our code we have tried to keep in mind and have stuck to an **MVC (Model View Controller)** format. We would try to incorporate this where possible but certainly not fuss too much if we could not bring it to work. This meant that everything would theoretically work a lot nicer later on, and maintaining the integrity of our application would be a lot less problematic in terms of any future changes, etc. We can safely say that we have maintained this MVC format through into our final application.

We thought, initially, that the application would need a login and that we had to manually set people to a task, at a specific time and state when they finish this task. We later realized that we were slightly on the wrong track and that everything should be set-up automatically. The aforementioned factors would instead be calculated in our Work Schedule. We took this on board and pursued development accordingly.

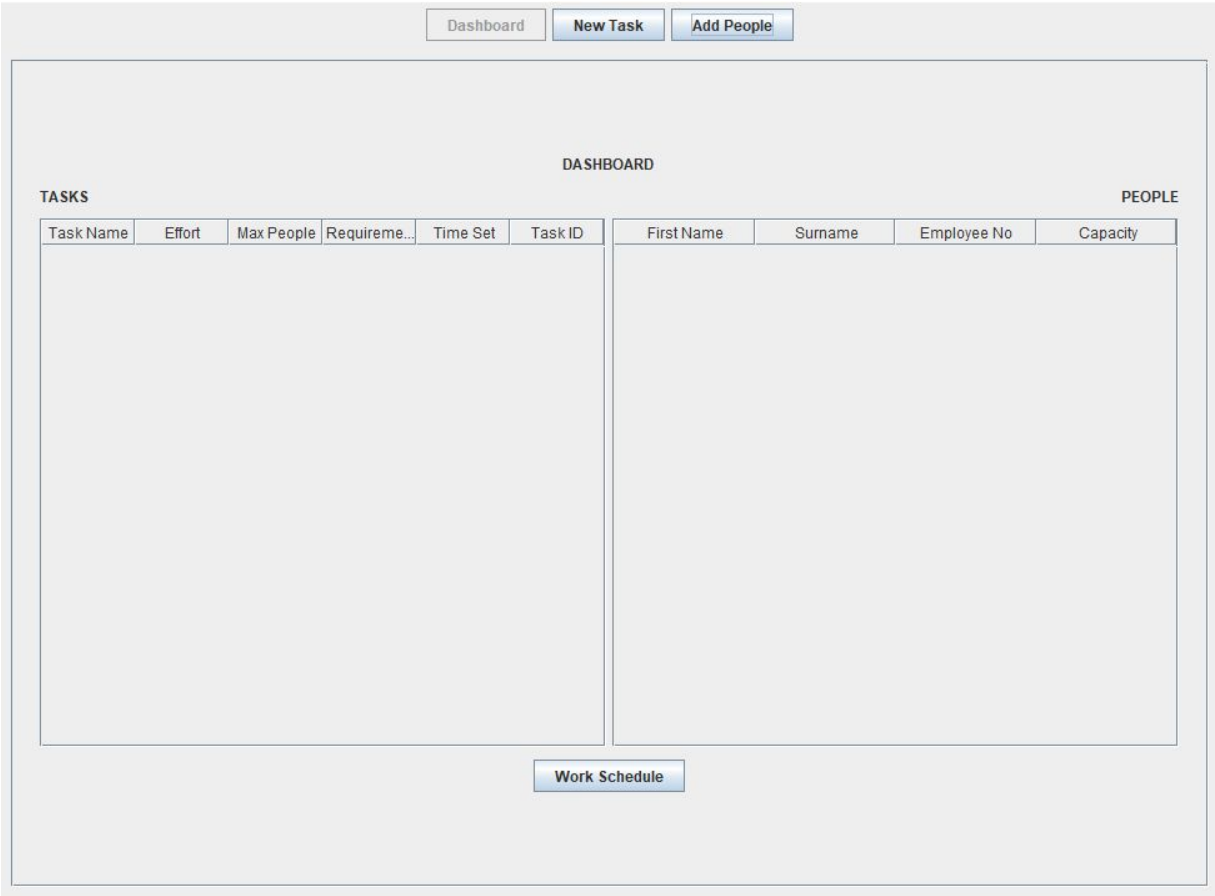
Graphical User Interface

One of the first steps in development of our application was our GUI, for which we use *Java Swing*. To be able to flick easily between pages, we have made the structure of this as convenient as possible for the user.

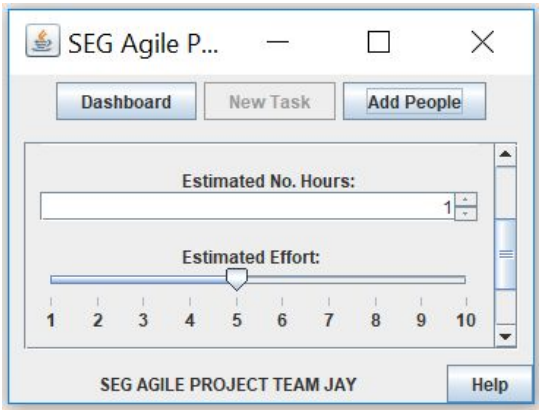
We have a MainGUI class containing each of five different panels (Panel1, Panel2, Panel3, Panel4 and Panel5, all of which are classes). Despite these panels all being GUIs in their own right, being contained within the MainGUI (which acts as a mainframe containing the other classes) allows switching between them to be made very easy.

The panels are a follows: Panel1 is our main dashboard, which shows two JTables: One of these tables presents all current tasks to the user, and the other does the same but for all people. The user can add new tasks and people into the database using the ‘New Task’ and ‘Add People’ buttons, seen at the top of the application, respectively. Should a user want to see a schedule calculated of their tasks, they can click on the ‘Work Schedule’ button in the middle of the dashboard.

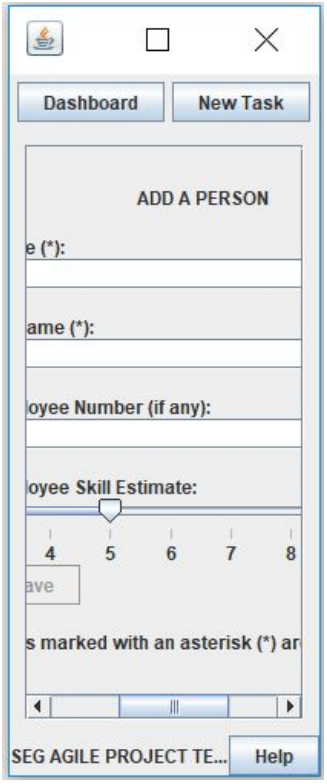
Panel2 is our page for adding new tasks, Panel3 is our page for adding new people, Panel4 is an instructions page and Panel5 is the interface for ‘Work Schedule’.



We have built each panel in our program so that content stays nicely centered and the application resizes nicely in this way. We use GridBagLayout to achieve this effect, along with a combination of other layouts. With this, we use a JScrollPane that appears when the user sizes the window below the size of the actual content - this way you can always view everything on the page, regardless of screen size. See below.



(JScrollPanes working as intended)



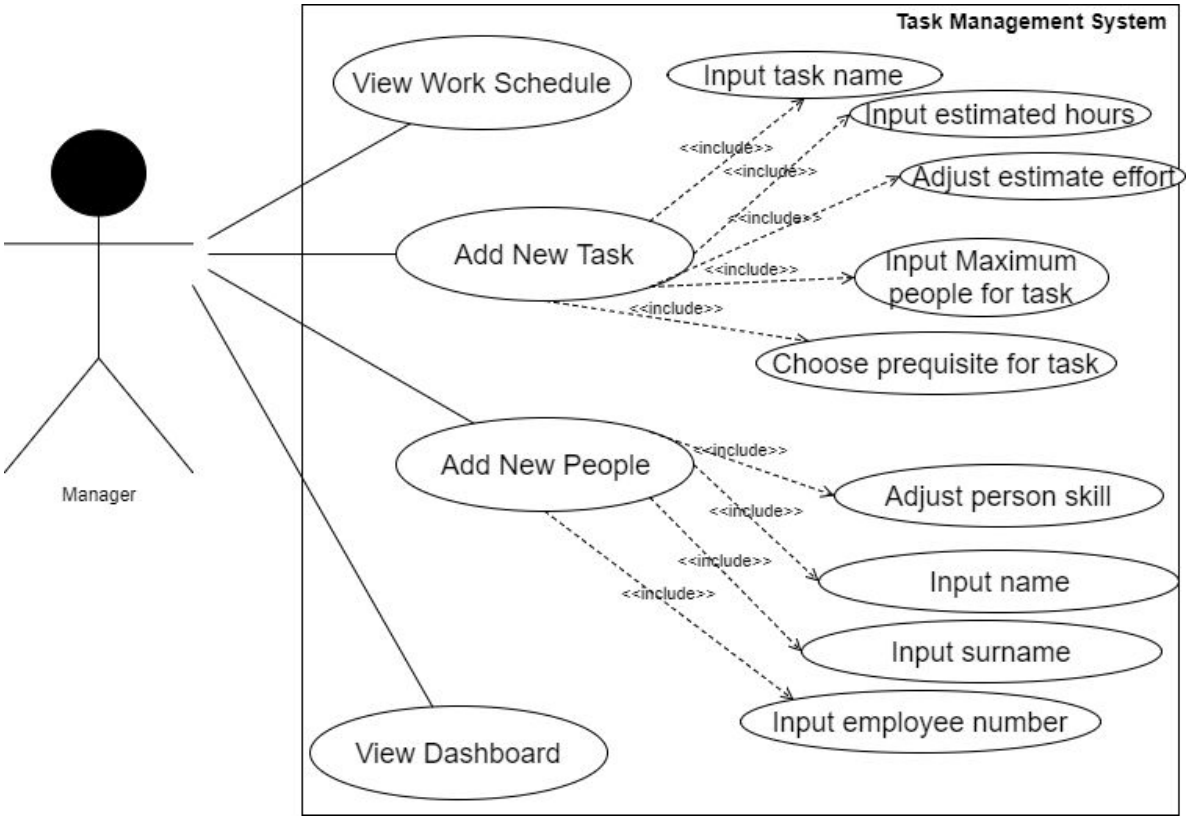
Final Product

In our program the user is able to create People to work on tasks, and assign them: an estimated ability in completing tasks, a first name, surname and an optional Employee Number (should they have one). Furthermore, you can create a Task with: its own estimated effort, name and estimated time in which to do it (this changes depending on the complexity of the task and the capacity or number of people working on it). A task is also given a maximum number of people that can work on it, as well as any additional requirements/dependencies that the task may require. These requirements are tasks that require completion before you can start the task you are currently creating. The user is able to view the dashboard to see all the data they have created, in tables, and they can view a Work Schedule that will automatically display the most efficient way a person(s) can do which task(s) at what times.

We have included an ‘instructions’ page in our application, although we hope we made our interface intuitive with labels where necessary. This page is an extra helper to users using our application.

Finally, we have a Kanban Board that we developed alongside development of our project. We have testing that we wrote, also as we developed our project, and we have a README file available for viewing in our GitHub repository.There are also various design or schedule elements we have made on paper over the duration of our project. For these, we have made a GitHub Wiki.

Use Case Diagram:



Architecture diagram: