

Software Engineering Plan of Work

Feb 16, 2018

<http://>

Project manager

Project dates

Feb 19, 2018 - May 4, 2018

Completion

0%

Tasks

22

Resources

9

This Gantt Chart is for the Home Automation group (group 15)

Tasks

2

Name	Begin date	End date
Report #2	2/19/18	3/9/18
Report #2 part 1	2/19/18	2/23/18

Interaction Diagrams

Do interaction diagrams for the use cases you elaborated ("fully dressed") in Report #1. You should do at least sequence diagrams, but you may do some other UML interaction diagrams, as well.

Describe what design principles you employ in the process of assigning responsibilities to objects. This can be done either as comment "bubbles" in the diagram, or in the caption of the diagram.

Read the UML textbook about interaction diagrams and Section 2.5.1 in the course lecture notes, in order to learn about good design principles

Here are examples of software tools for UML diagramming available for free download

Project Management (described in Section 7 below) and References (described in Section 8 below)

Tasks

Name	Begin date	End date
Report #2 part 2	2/26/18	3/2/18

Tasks

Name	Begin date	End date
Part 2:		
<p><i>Class Diagram and Interface Specification</i></p> <p><i>Class Diagram</i></p> <p>Show all classes and their associations. Only indicate visibilities of attributes and operations; full details about the types and signatures should be provided in the next item.</p> <p>If you cannot fit the class diagram on one page, or it looks too cluttered, create one “overview” class diagram showing all classes and their relationships, but for each class show only a single compartment with the class name (leave out attributes and operations).</p> <p>Then on subsequent pages show partial class diagrams, with three compartments and all the attributes and operations of a class. Make sure to indicate in diagrams and describe in text how partial diagrams fit into the overall class diagram.</p> <p><i>Data Types and Operation Signatures</i></p> <p>Independently of the class diagram, write down class specification in UML notation. For every class, specify data types of all attributes and operation signatures.</p> <p>Define the meaning of each class, operation, and attribute in plain language.</p> <p><i>Traceability Matrix</i></p> <p>Show how your classes evolved from your domain concepts. Provide explanations for modified names or multiple classes that evolved from a single concept—providing only a matrix with checkmarks is not enough!</p> <p>Note that if your system has many classes, the matrix may become difficult to read. Alternatively, you may list line-by-line, all your domain concepts and explain which classes were derived from each concept and why.</p> <p><i>System Architecture and System Design</i></p> <p><i>Architectural Styles</i></p> <p>Describe the architectural styles used in your design (Google search software architectural styles example)</p> <p><i>Identifying Subsystems</i></p> <p>Draw and describe UML package diagram of subsystems in your system (See the UML textbook)</p> <p><i>Mapping Subsystems to Hardware</i></p> <p>Does your system need to run on multiple computers? For example, you may have a client (web browser) and a server (web server) subsystems, running on different machines.</p> <p>If NO, skip to the next item;</p> <p>If YES, what subsystems run on which machine?</p> <p><i>Persistent Data Storage</i></p> <p>Does your system need to save data that need to outlive a single execution of the system?</p> <p>If NO, skip to the next item;</p> <p>If YES, identify the persistent objects and select the storage management strategy, e.g., flat files, relational database, etc.</p> <p>Attach the description of the file format and/or database schema (format of database tables, printed by the command description)</p> <p><i>Network Protocol</i></p> <p>If your system runs on a single machine, this question probably does NOT apply; skip to the next item;</p> <p>Otherwise, which communication protocol do you use, e.g., plain Java sockets, Java RMI, Java JDBC, HTTP, etc. Explain why you made your specific choice.</p> <p>If you are using plain sockets, attach the description of the communication protocol (types of messages, message format, etc.).</p> <p><i>Global Control Flow</i></p> <p><i>Execution orderness</i>: Is your system procedure-driven and executes in a “linear” fashion, where every user every time has to go through the same steps, or is it an event-driven system that waits in a loop for events, and every user can generate the actions in a different order?</p> <p><i>Time dependency</i>: Is there any timers in your system?</p> <p>Is your system of event-response type, with no concern for real time, or is it a real-time system? If it is real-time, is it periodic, and what are the time constraints for each period?</p> <p><i>Concurrency</i>: Does your system use multiple threads?</p> <p>If NO, skip to the next item;</p> <p>If YES, identify the objects that have separate threads of control and describe any synchronization between the threads?</p> <p><i>Hardware Requirements</i></p> <p>What system does resources your system depend upon? Examples are, screen display, disk storage,</p>		

Tasks

Name	Begin date	End date
<i>communication network, or you may be accessing some special sensor/instrument. Describe exact requirements for these resources in order for your system to run. For example, you need color display, with minimum resolution of 640 × 480 pixels; minimum of 2 Gbytes hard disk space; minimum network bandwidth 56 Kbps.</i>		
<i>Project Management (described in Section 7 below) and References (described in Section 8 below)</i>		

Tasks

Name	Begin date	End date
Report #2 part 3	3/5/18	3/9/18

Tasks

Name	Begin date	End date
<p>Algorithms and Data Structures (if applicable)</p> <p>Algorithms</p> <p>Describe the algorithms that implement mathematical models from your Report #1. Does your system use any other complex algorithms? For example, when computing a motion trajectory for an animate figure in a game, you may use some numerical or computer-graphics algorithms. Or, when assessing stock market movements, you may be using statistical algorithms.</p> <p>If NO, skip to the next item;</p> <p>If YES, describe your algorithms. For example, for the animate figure example above, will the path coordinates be precomputed and stored in a look-up table or will they be computed using a spline interpolation algorithm.</p> <p>It is a good idea to use activity diagrams to describe the algorithm design.</p> <p>Data Structures</p> <p>Does your system use any complex data structures, such as arrays, linked lists, hash tables, or trees?</p> <p>If NO, skip to the next item;</p> <p>If YES, what criteria you used in deciding what data structure to use, e.g., performance vs. flexibility?</p> <p>User Interface Design and Implementation</p> <p>Describe whether and how you modified and implemented the initial screen mock-ups developed for Report #1. Comment only on significant changes in your user interface, those that reduce (or increase) the user effort. Changes of colors or styles are less important and should be omitted from your report.</p> <p>The textbook does not deal much with the GUI design. Excellent guidelines for GUI design can be found here: Sun Microsystems, Inc. Java Look and Feel Design Guidelines. Mountain View, CA, 1999. Available at: http://java.sun.com/products/jlfd2/book/</p> <p>"Ease-of-use" is generally considered a key characteristic of user interface. "Ease-of-use" should not be confused with a flashy interface, with lots of colors, picture, graphics, etc. On the contrary, you should avoid flashy user interfaces. "Ease-of-use" means that interface is intuitive, easy to understand and operate, without having to ask many questions or read voluminous documentation. A minimal user interface that is well organized should be sufficient. You already considered the user effort as part of Report #1, and here you should strive to minimize the user effort, thus maximizing the "ease-of-use".</p> <p>Design of Tests</p> <p>Note that for this report you are just designing your tests; you will program and run those tests as part of work for your first demo, see the list here.</p> <p>List and describe the test cases that will be programmed and used for unit testing of your software.</p> <p>Discuss the test coverage of your tests.</p> <p>Describe your Integration Testing strategy and plans on how you will conduct it.</p> <p>Describe also your plans for testing any algorithms, non-functional requirements, or user interface requirements that you might have stated in your Report #1.</p> <p>Project Management and Plan of Work</p> <p>Merging the Contributions from Individual Team Members</p> <p>Compiling the final copy of the report from everyone's work, ensuring consistency, uniform formatting and appearance.</p> <p>Describe what issues were encountered and how they were tackled.</p> <p>Project Coordination and Progress Report</p> <p>What use cases have been implemented?</p> <p>What is already functional, what is currently being tackled?</p> <p>List and describe other relevant project management activities.</p> <p>Plan of Work</p> <p>List the projected milestones and dates by which you plan to accomplish them. Preferably, you should use Gantt charts for planning and scheduling your project.</p> <p>Breakdown of Responsibilities</p> <p>List the names of modules and classes that each team member is currently responsible for developing, coding, and testing</p> <p>Who will coordinate the integration?</p> <p>Who will perform and integration testing? (The assumption is that the unit testing will be done for each unit by the student who developed that unit.)</p> <p>References</p> <p>The list of references should contain exact references and URLs of any material that is used in the project and does not come from the textbook.</p>		

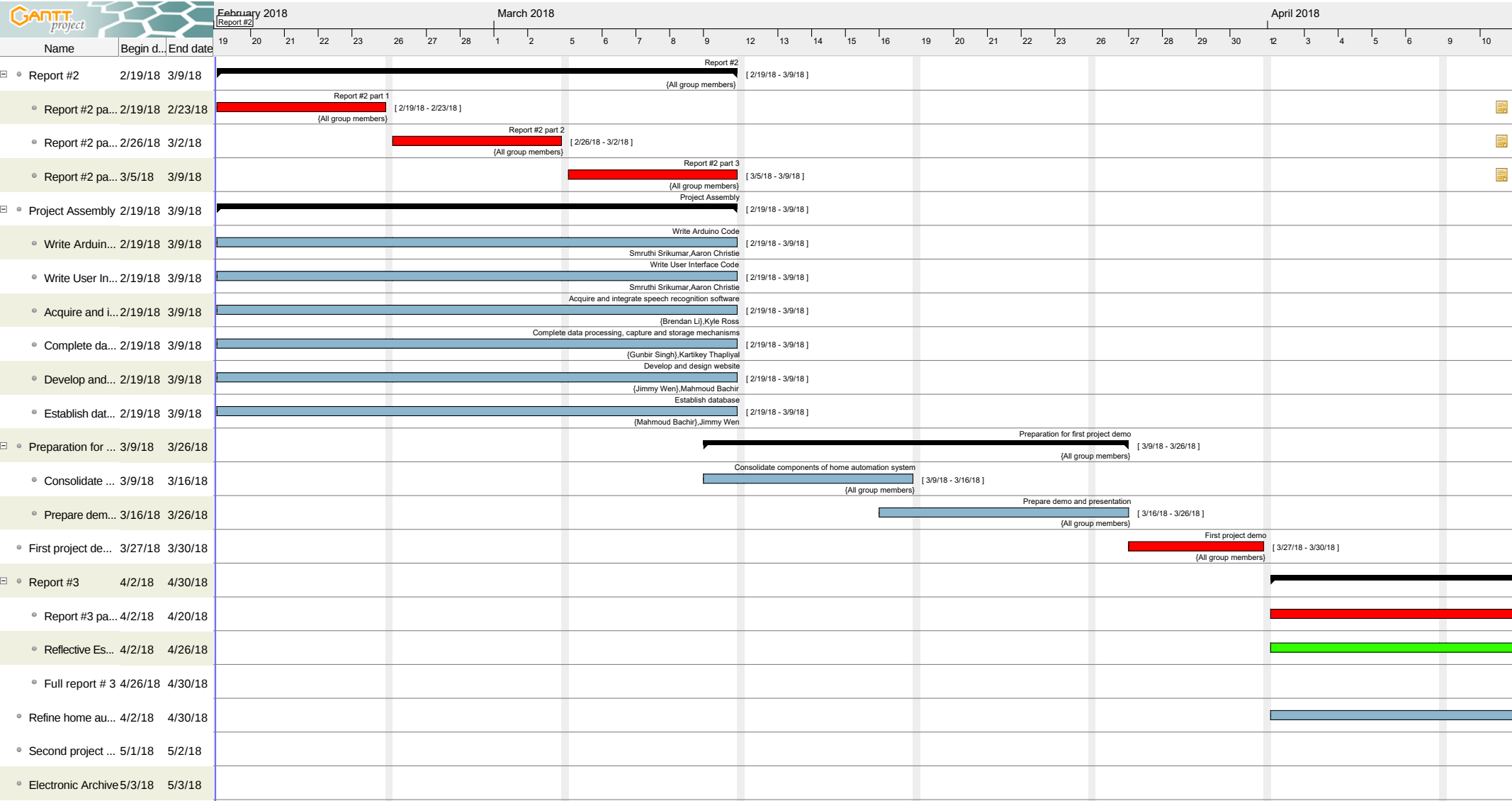
Tasks

Name	Begin date	End date
Project Assembly	2/19/18	3/9/18
Write Arduino Code	2/19/18	3/9/18
Write User Interface Code	2/19/18	3/9/18
Acquire and integrate speech recognition software	2/19/18	3/9/18
Complete data processing, capture and storage mechanisms	2/19/18	3/9/18
Develop and design website	2/19/18	3/9/18
Establish database	2/19/18	3/9/18
Preparation for first project demo	3/9/18	3/26/18
Consolidate components of home automation system	3/9/18	3/16/18
Prepare demo and presentation	3/16/18	3/26/18
First project demo	3/27/18	3/30/18
Report #3	4/2/18	4/30/18
Report #3 part 1	4/2/18	4/20/18
Reflective Essay	4/2/18	4/26/18
Full report # 3	4/26/18	4/30/18
Refine home automation system	4/2/18	4/30/18
Second project demo	5/1/18	5/2/18
Electronic Archive	5/3/18	5/3/18

Resources

Name	Default role
All group members	undefined
Mahmoud Bachir	Project lead
Smruthi Srikumar	User Interface Designer
Aaron Christie	Arduino coder
Jimmy Wen	Web developer
Gunbir Singh	Data manager
Kartikey Thapliyal	Data manager
Brendan Li	Speech recognition developer
Kyle Ross	Speech recognition developer

Gantt Chart



Resources Chart

