Software Project Management: Graphical Password Strategy

SOFE: Software Project Management

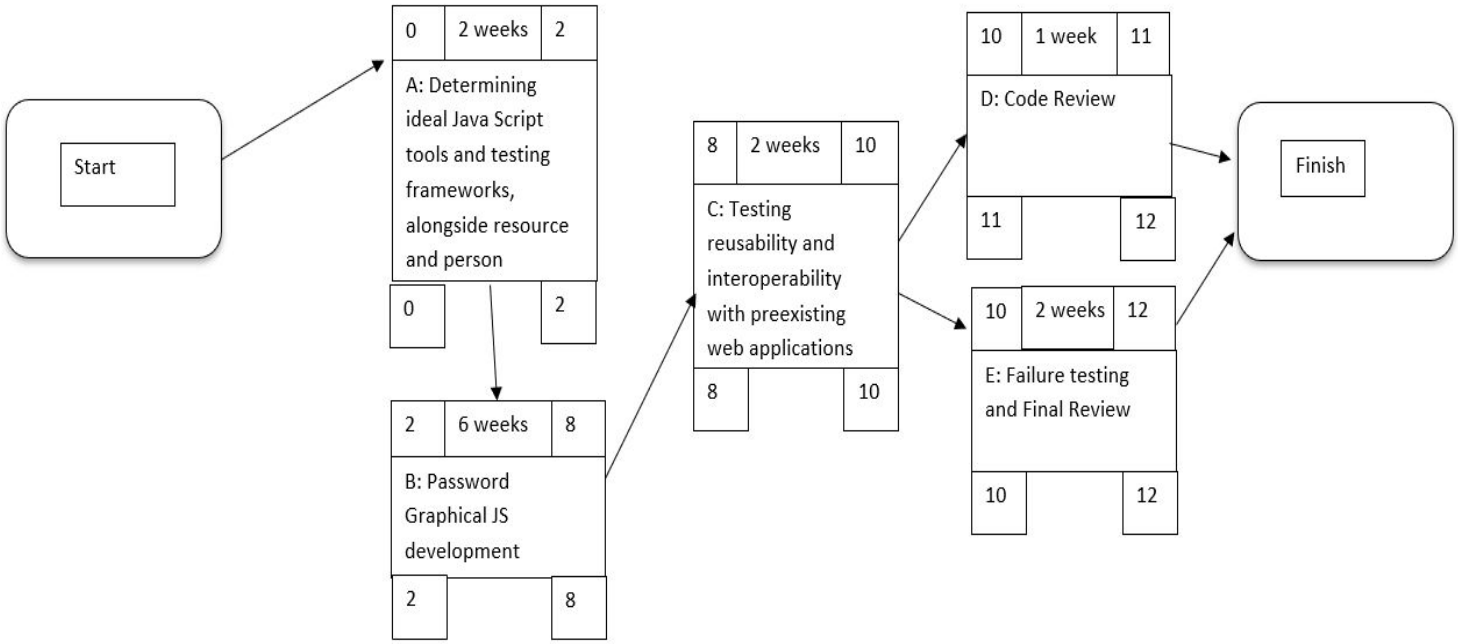
Lab 4

Hamza Farhat Ali - 100657374

Mohammad Minhal Syed - 100618744

Marvin Vuu - 100658622

Risks with Activities:



Referring to the activity diagram above, these are 4 different types of risks that may be identified during the project. The team has decided to add tasks for counter measure to risks, activity diagram is not referenced strictly:

Risk 1: Customer is not satisfied with the final graphical strategy web application.
Countermeasure: To hold an open beta version of the software to be released to the stakeholders as a test. This allows for feedback and will be done as tasks as activity D and E of the activity diagram. Due to one of the measures of success of this project is determined by the user and positive feedback, it is very important for this risk to be planned and have important tests for a perfect final version.

Risk 2: The amount of time needed to complete each task is underestimated.
Countermeasure: The activities that are set above allow for change as the time is always overestimated. This risk planning occurs for tasks that might take longer than usual. For example, the devops team might not be able to include all functionalities within the program. Therefore, multiple deadlines for each sub task within a task is better for progress checking and completion, so that this risk does not occur. These deadlines within an activity will be happening throughout all activities as the product needs to be delivered on time. This risk mitigation will be done through once a week meetings between developers to allow for errors and potential drawbacks to not occur.

Risk 3: Requirements change during the development phase.
Countermeasure: Since the project objectives were set as below:

1. Create a fully functional Graphical Password that works with only the correct images and sequence determined by the user
2. Show a certain amount of images determined by the user when deciding password, limit of 100 images.
3. Let the user input the correct order of images to select.
4. Allow users to upload their own images or choose a category of images to create a password.
5. Option of having one password for all accounts.

If the stakeholder needs to add another objective or change a current one, different alternatives can be done. Stakeholder meetings can be set up during the development phase to re-outline what requirements are needed, for example only 30 images limit instead of 100 images. Also, if the senior software developer in the devops team decides to use "Off the shelf" tools in order to achieve this web application, it must be discussed in meetings. This risk mitigation between the team and the customer allows for proper updates and more chances for feedback, following an agile development approach.

Risk 4: Workload is too heavy for employees.
Countermeasure: Look for different ways to get the job done through brainstorming sessions to simplify/eliminate certain tasks. As well as, allocate resources properly to make sure each activity will have enough employees to finish tasks on time, and must not be under or over staffed. Having a safe atmosphere at work to make it easier for employees to ask for help.

Risk Analysis:

The scale used is 1-10, likelihood and impact are multiplied in order to get the total risk.

	Likelihood	**Impact**	**Risk**
Risk 1	7	7	49
Risk 2	5	8	40
Risk 3	5	5	25
Risk 4	3	5	15

Resource Allocation :

Devops Team: Responsible for determining the ideal tools & testing framework, as well as developing the code for the project. The software being created is a web application as this page can be adding to pre existing online portals to login. This will be done by using off the shelf software such as Microsoft Azure. This will allow us to save time as it is already developed and tested so there is no need to worry about backend issues or security breaches. As well as the front end will be created using React and React Native. This modern JavaScript framework will allow the team to create cross platform applications for Chrome, Internet explorer, Mozilla etc. The graphical password strategy can also be added as user login in mobile applications, by building on React and using React Native. This team will be in charge of the development phase.

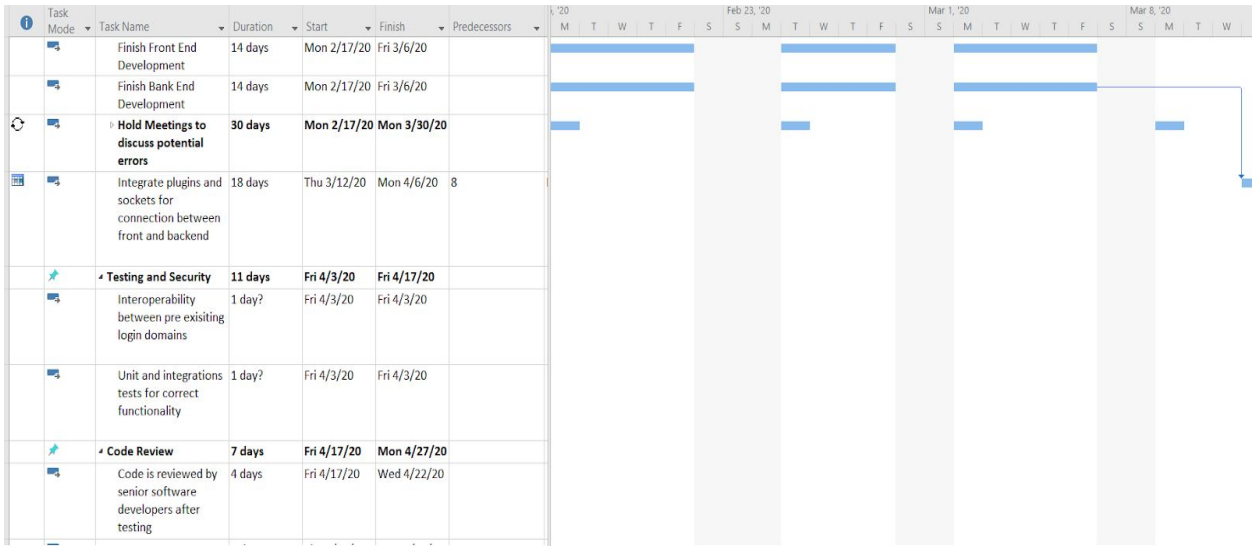
Quality Assurance Team: Responsible for monitoring, analyzing and testing the software development in order to ensure the quality of the product that is being produced. They will be creating test cases and test plans to ensure the product is well functioning at completion while minimizing and catching bugs. The tool that allows JS testing is Mocha, to write unit tests and integration tests. In addition, the Selenium Web Driver will allow for different automated tests to occur for user clicks on the password. Finally, Gerrit will be used alongside github to allow for testing code with multiple testers.

Security Team: Responsible for ensuring the security and upkeep of the software once it has been produced, so we are not vulnerable to attacks on sensitive information clients entrust us with. This will mostly be done by stress testing the software, as well as encrypting the data.

Project Management: Responsible for determining and defining the scope of the project as well as creating the activity diagram to plan out each part as well as the duration. They develop and manage a detailed project schedule and utilize industry standards throughout the project execution. Microsoft Project was used by the team to allow for easier resource allocation and tasks assigned to each team.

Gantt Chart:

Refer to Microsoft Project file for full chart



Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Finish Front End Development	14 days	Mon 2/17/20	Fri 3/6/20	
	Finish Bank End Development	14 days	Mon 2/17/20	Fri 3/6/20	
	Hold Meetings to discuss potential errors	**30 days**	**Mon 2/17/20**	**Mon 3/30/20**	
	Integrate plugins and sockets for connection between front and backend	18 days	Thu 3/12/20	Mon 4/6/20	8
	Testing and Security	**11 days**	**Fri 4/3/20**	**Fri 4/17/20**	
	Interoperability between pre exisiting login domains	1 day?	Fri 4/3/20	Fri 4/3/20	
	Unit and integrations tests for correct functionality	1 day?	Fri 4/3/20	Fri 4/3/20	
	Code Review	**7 days**	**Fri 4/17/20**	**Mon 4/27/20**	
	Code is reviewed by senior software developers after testing	4 days	Fri 4/17/20	Wed 4/22/20	