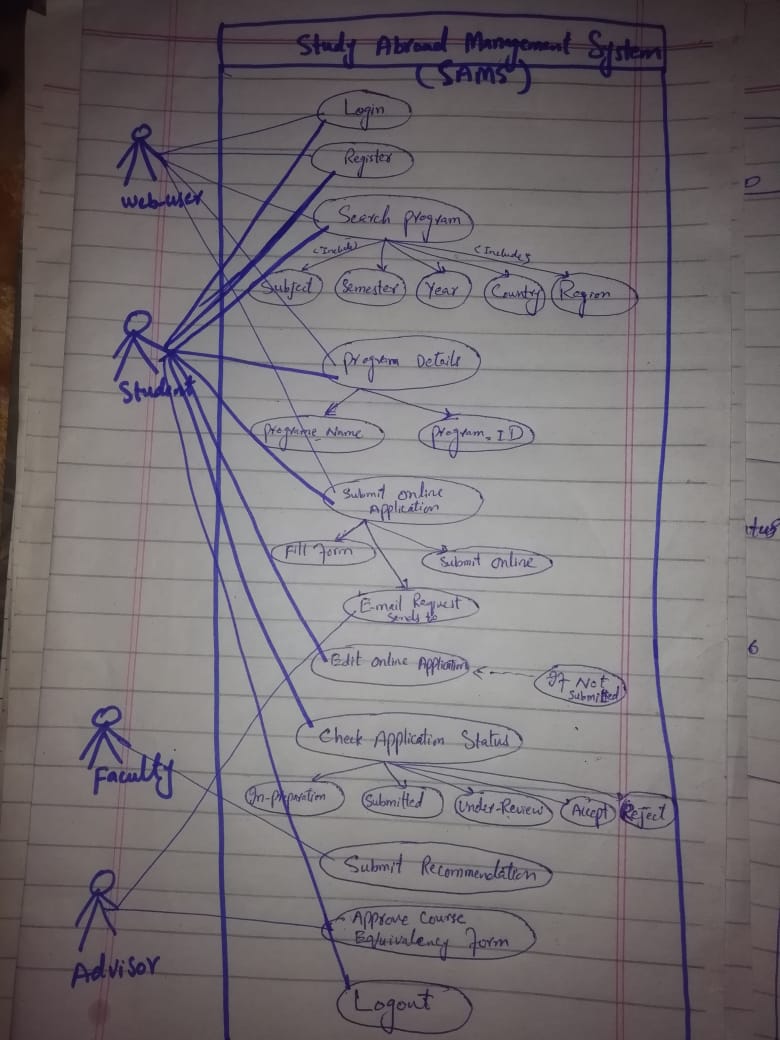
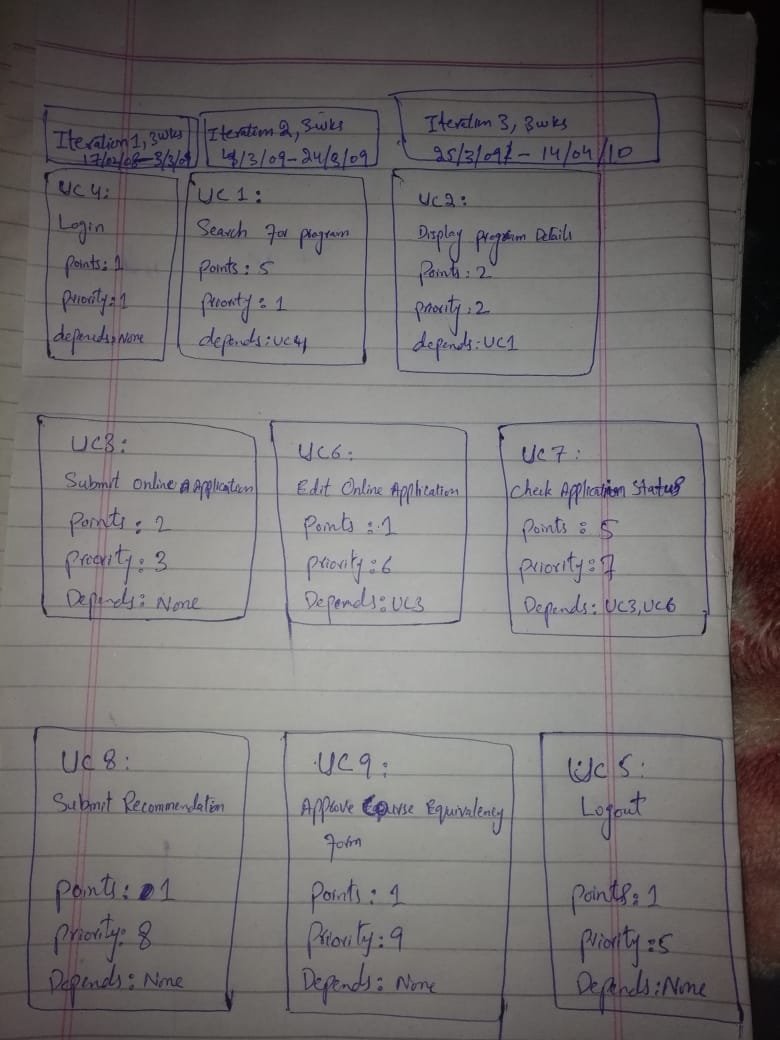
**Use Case:**

****

**1. Identify the dependencies among the nine use cases for the Study Abroad application:**

****

**2. Based on the estimates recorded in your lab results, use Agile planning to plan iterations with use cases. Make necessary assumptions when applying story points to use cases. Assume a staff of three, an app designer, a developer, and a tester. Assume that the team can complete 3 points per week, which is the equivalent of 120 hours of effort per week.**

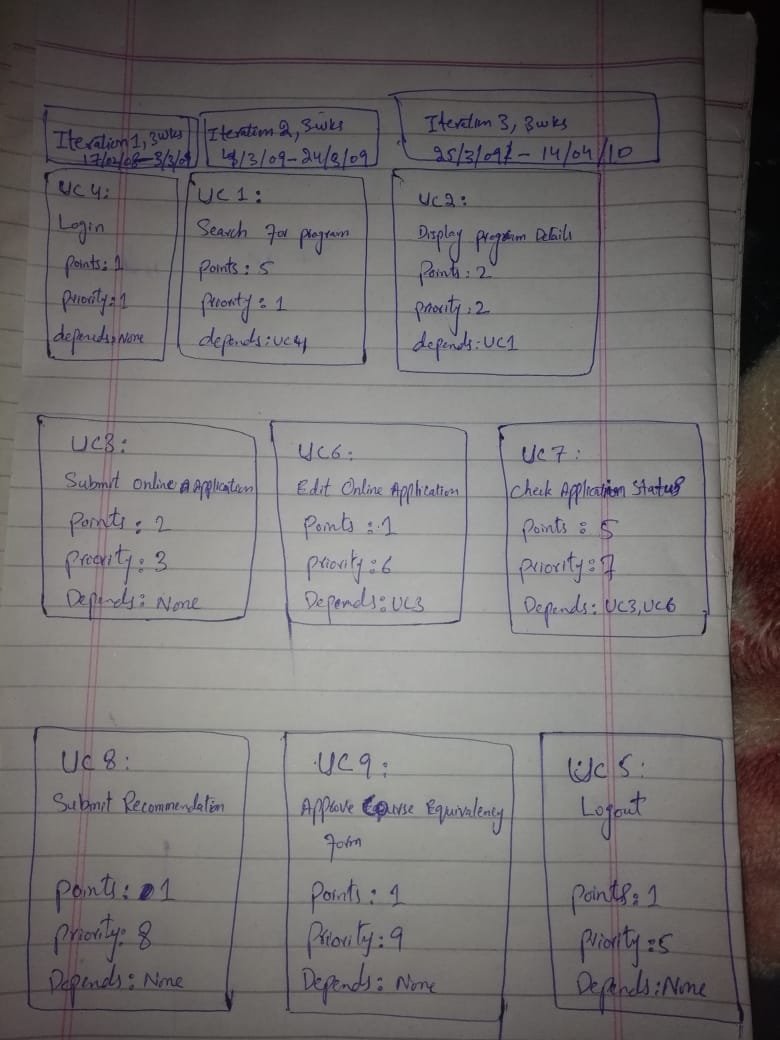
**Sprint Planning Process:**

1. **Do a retrospective meeting** to discuss the previous sprints and lessons learned.
2. **Run a sprint planning meeting**to analyze the release plan and update it according to velocity in recent sprints, changes to priorities, new features, or idle time that wasn’t planned for in the release.
3. **Make sure user stories are detailed enough**to work on. Elaborate on tasks that are not well defined, to avoid surprises.
4. **Break down user stories into specific tasks**. For example, the user story “view tasks assigned to me” can be broken up into UX design of a “my tasks screen”, back-end implementation, and front-end development of the interface. Keep size of tasks small, no more than one work day.
5. **Assign tasks to team members**and confirm that they are committed to performing them. In the agile/scrum framework this is done by the Scrum Master.
6. **Write the tasks on (physical) sticky cards**and hang them up on a large board visible to the entire team. All the user stories in the current sprint should be up on the board.
7. **Track progress of all the tasks**on a grid, by recording who is responsible for completing each task, estimated time to complete it, remaining hours, and actual hours used. This time tracking should be updated by all team members and visible to everyone.
8. **Track velocity using a burndown chart.**During the sprint, use the team’s time tracking to calculate a chart showing the number of tasks or hours remaining, vs. the plan. The slope of the burndown chart shows if we are on schedule, ahead, or behind schedule.

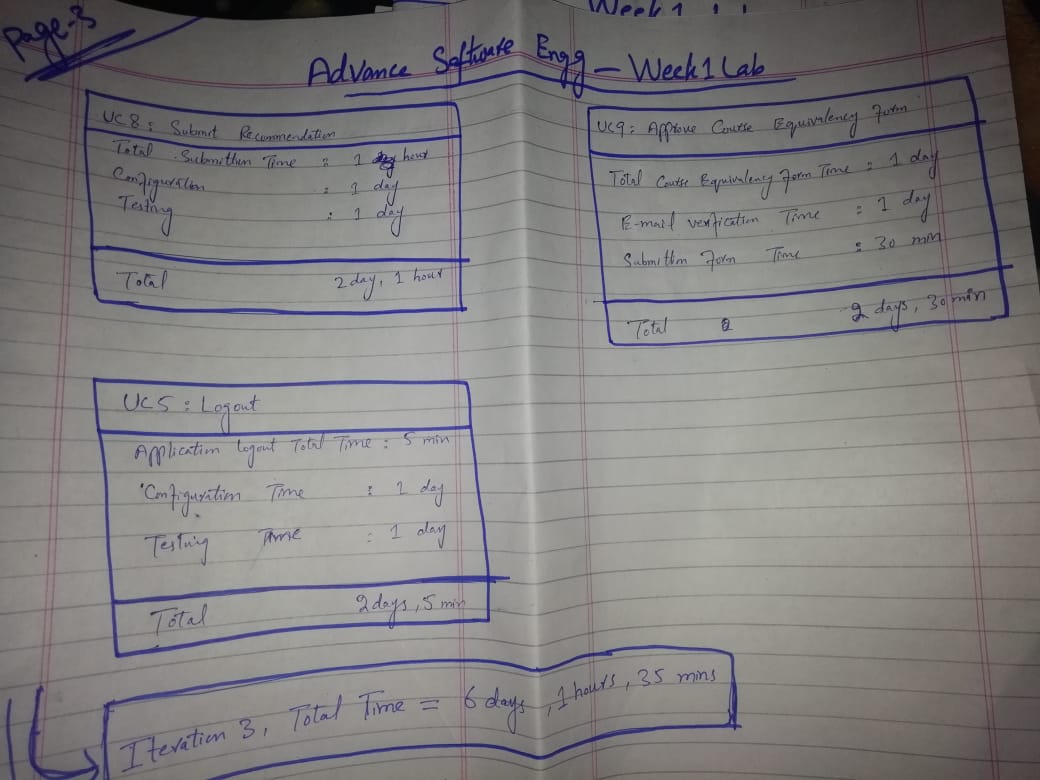
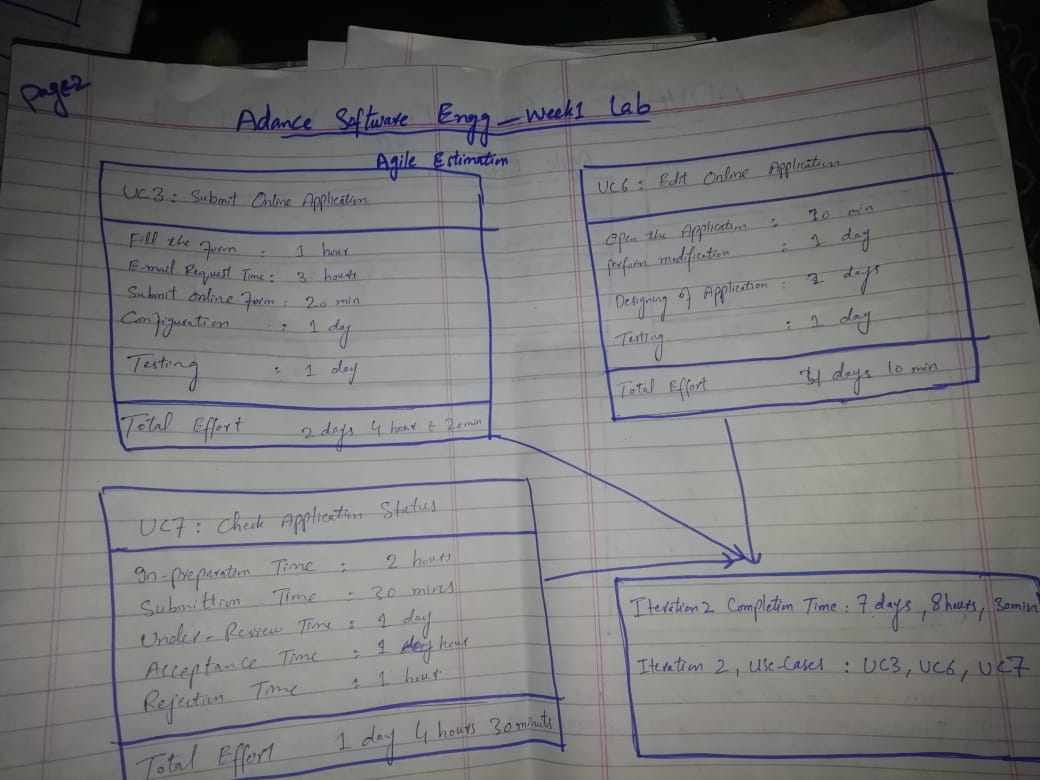
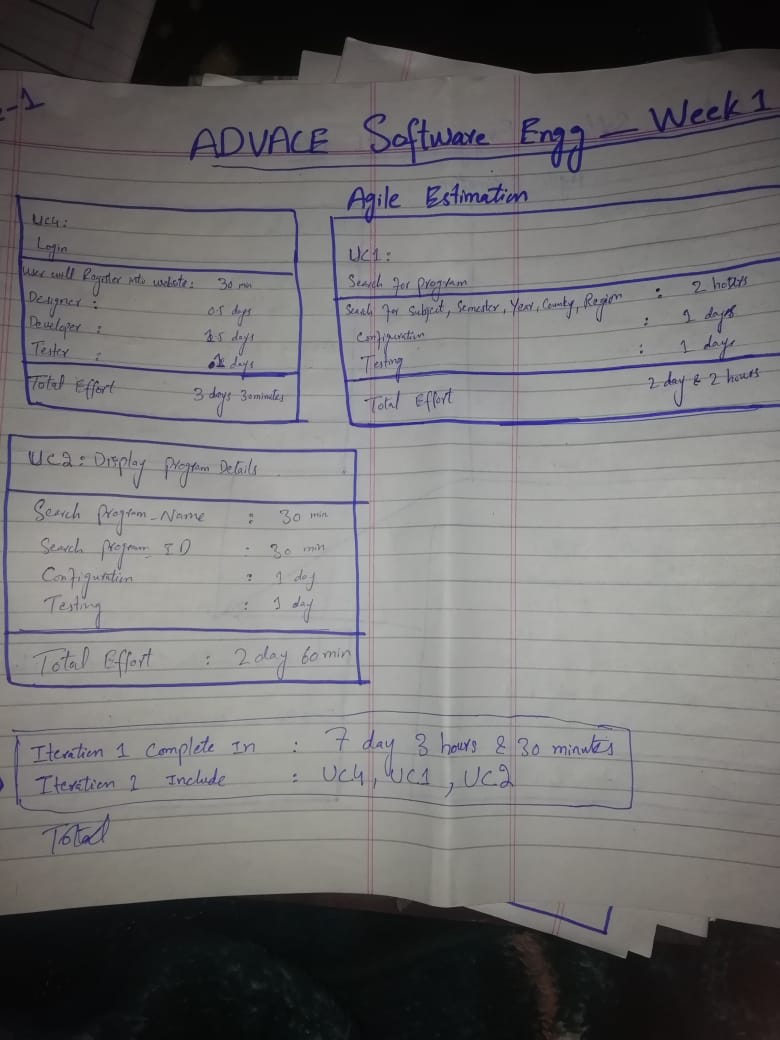
**Example link is below:**

[**Project issue tracker1 example.xlsx**](Project%20issue%20tracker1%20example.xlsx)

**3. The project must be completed in 3, one-month iterations.**

****

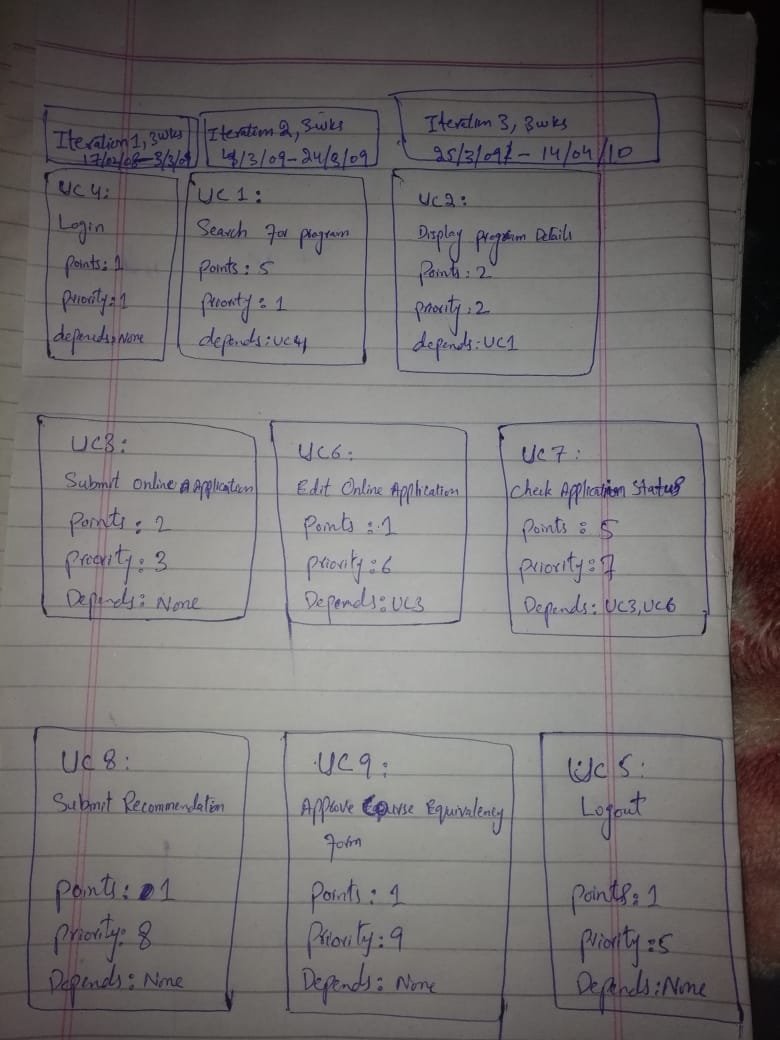
**4. Submit an image containing a corkboard with story cards, or a grid of post-it notes, or an Excel spreadsheet may also be submitted. Show a grid with the iterations on the top row and the use cases in the columns below.**

****

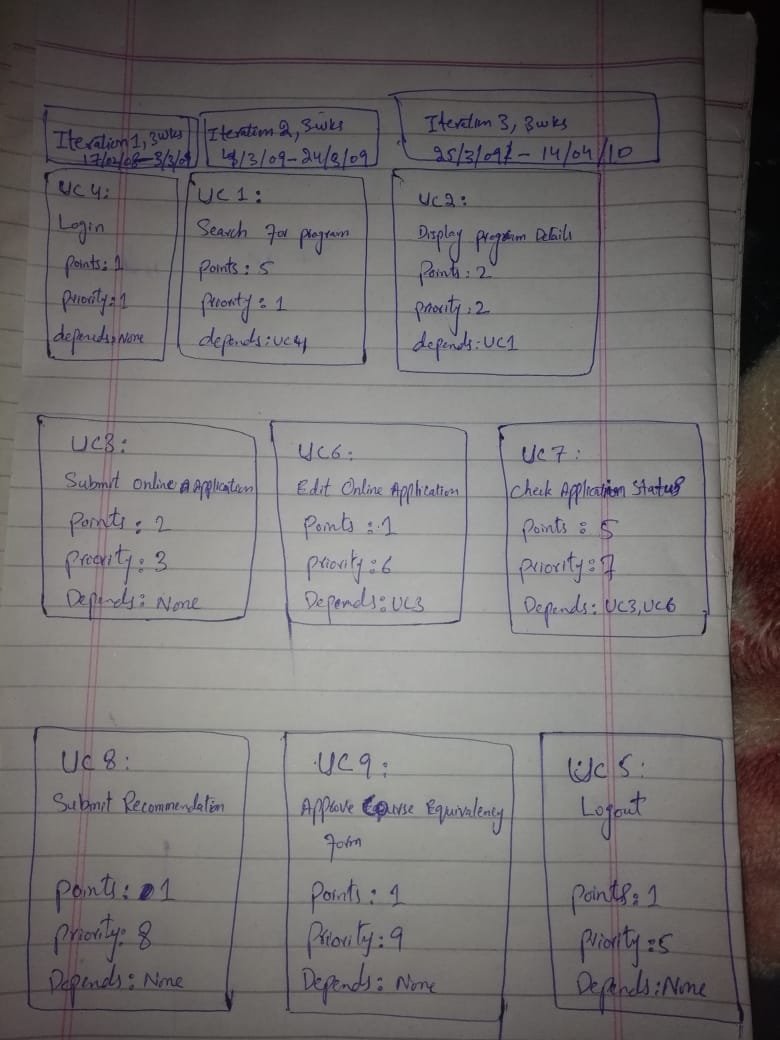
**Excel chart link is below:**

[**Agile Gantt chart1.xlsx**](Agile%20Gantt%20chart1.xlsx)

**5. On the top row, number each iteration, show the number of weeks the iteration will last, and show the beginning and ending dates**

****

**6. In each column, show the use cases that will be completed during the iteration: a. Show the use case number b. Show the number of points c. Show the priority of the use case d. Show other use cases on which the use case depends**

****

**7. If you find that the use cases cannot be completed within three iterations then state it and explain why. If you find that more staff is needed to complete the use cases in three onemonth iterations, state it. If you think that another iteration should be added, state it. If you think the project could be completed in less than three one-month iterations, state it.**

Use Cases are completed within iterations and Staff is managed in all iterations and no extra staff is required. No further iteration is needed. If our staff will more experienced and our testing and configuration will strong then we can complete project less than three one-month iterations.

8. **Identify top five possible risks for the SAMS project and develop risk resolution measures for them. Briefly justify your solution**

**Risk 1: Inherent Schedule Flaws**

**Explanation:** Software development, given the intangible nature and uniqueness of software, is inherently difficult to estimate and schedule.

**Waltzing…Solution:** Get the team more involved in planning and estimating. Get early feedback and address slips directly with stakeholders.

**Risk 2: Requirements Inflation**

**Explanation:** As the project progresses more and more features that were not identified at the beginning of the project emerge that threaten estimates and timelines.

**Waltzing…Solution:** Constant involvement of customers and developers.

**Risk 3: Employee Turnover**

**Explanation:** Key personnel leave the project taking critical information with them that significantly delays or derails the project.

**Waltzing…Solution:** Increased collaboration and information sharing on the team.

**Risk 4: Specification Breakdown**

**Explanation:** When coding and integration begin it becomes apparent that the specification is incomplete or contains conflicting requirements.

**Waltzing…Solution:** Use a dedicated Product Manager to make critical trade off decisions.

**Risk 5: Poor Productivity**

**Explanation:** Given long project timelines, the sense of urgency to work in earnest is often absent resulting to time lost in early project stages that can never be regained.

**Waltzing…Solution:** Short iterations, right people on team, coaching and team development.