

## CSC 648/848 SFSU 2024 Milestone 3

### Review of functionality, UI, SW and planning for final product delivery

Includes Checklist for instructors and teams (Appendix I and II)

#### Objective and Overview

The objectives of Milestone 3 are to:

- **Define exactly what product you are delivering.** We will come to agreement on what the final application is going to look like in terms of functionality, especially which functional requirements are priority 1 (P1). This will be your commitment to the instructor/client to delivery by M5.
- **Ensure software development is on track.** We will verify that the all the SW components are installed and integrated and that most major functions work.
- **Provide feedback on all major UI screens and functionality.** This will be done through a review of the so-called horizontal or UI prototype.
- **Check software architecture** by reviewing the database and overall design at a high level UML class diagram and sequence diagram.
- **Check all algorithms** like search or machine learning component, whatever major algorithm in your application will be checked.
- **Identify and address all technical risks**
- **Ensure effective teamwork** by verifying that all team members have started implementation of their portion of the project.
- **Ensure software development is effective** by verifying your team's collaboration practices based on github branch policy and github review policy.

Milestone 3 will be in the form of two-part review:

1. **Part 1 – each team presents to Prof. Song: Review of functionality, UI and general project status** will be done during the meeting (with each team) of ~12 minutes. The meeting will happen during the class on the deadline day. In order to be efficient, teams must observe strict schedule and come fully prepared. For this, please prepare Appendix I (item 2,4) before the M3 demo. And please finalize Appendix I after the meeting based on Prof. Song's feedback. The Appendix 1 should be submitted to your M3 folder for review (Due : on the next day of M3 demo).
2. **Part 2: SW review (in-emails):** github usage, branch organization, code review practices by TA after Part 1 review.

M3 will be graded, and the feedback will be given if any. The teams will meet to analyze feedback and revise the design and implementation accordingly as well as definition of P1 features. After M3, the teams will have “**feature freeze**” e.g. the teams must focus on intense implementation of P1 features.

## **Part 1 Review – Functionality and UI feedback and general project status**

### **What to bring to the meeting**

Each team **must** prepare your **Product prototype** to the Milestone 3 Part 1 meeting:

- **Product prototype** has limited functionality
- You have to demonstrate the 5~6 key P1 functionalities for your product prototype for the meeting. For the key functionalities, you should connect back-end and front-end.
- The product prototype should provide UI implementation of 5~6 key P1 functionality.
  - The UI implementation should follow UX flow defined in M2.
- The current version of your SW should run on deployment server.

On the part1 meeting, the instructor will let each team to demonstrate major functionalities on real-time using your SW and will give you feedback. **You are requested to appoint a scribe who collects the feedback. Use Appendix I as a template to record feedback.**

**After the M3 Part 1 meeting (recommended to do it immediately after the meeting):** Team has to meet, analyze meeting feedback and revise M3 doc (Appendix 1), design and implementation as necessary. Team also must finalize P1 set of features. The instructor feedback as well as finalized P1 list **MUST** be written down using template as in Appendix I. You will submit it with M3 folder, by the next day of M3 part 1 meeting.

**Part 2 review: SW review – to be done by TA after Part I review, by accessing your github repo. See Appendix II for details**

## **Appendix I – Rubrics and checklist for Part 1 Milestone 3 review: Project Status and UI Review.**

**Section: 01   Team: 02      Date: 11/04/2024**

**Number of students present: 6**

### ***1. UI and functionality feedback (P1 functions only)***

During the meeting, students will demonstrate to run your SW from deployment server:

- Test 5~6 P1 features
- Show UI and usability: adherence to the feedback on UI mockup at M2, layout, flow, clarity, functionality etc.

Instructor will

- Check functionality and record issues/observe bugs
- Share comments on key UI and functional implementation
- Verify enough web pages are implemented and connected
- Verify Performance of web page

**Students must** record the instructor's comment in the below. Then the team should meet to analyze feedback, prioritize and revise and plan to implement changes accordingly. **Note that immediately after the review meeting with the instructor, the team must finalize P1 set of features and focus only on those from then on.**

- Instructor's comments on UI/functionality for your demo (should be during the class of M3 demo)
  - 1. The game review functionality should be implemented by milestone 4.
  - 2. We need to establish a better API interface document, so frontend does not need to understand backend.
  - 3. We should use Postman to test backend fetching.
  - 4. We need to design a UI for recommended games.
  - 5. We should delete some unnecessary github branches to avoid confusion.
- Your Plan for the comments

This week:

- As team leader I am going to prepare a more detailed API interface document and tutorial then share it on our Discord and in our Milestone 3 folder.
- I will also prepare a tutorial for how to use Postman and share on our Discord

- Our Github master will conduct an audit of the Github branches

Week of 11/11

- Kayla and Joyce will design recommended games UI
- Backend for game reviews should be completed
- Remaining backend bugs should be worked out and tested with Postman

Week of 11/18

- Frontend team will complete UI for game reviews
- Finish connecting frontend UI to backend

Week of 11/25

- Expanding test cases and working out bugs
- Usability testing and documentation

## ***2. List of P1 features committed for delivery***

Write down the candidate list of features before the demo and, verbally explain it during the meeting if time is allowed. Based on the instructor's review comment, finalize your committed features. Once you commit at M3, you can not change during the rest of the semester. You should implement the committed features by M5.

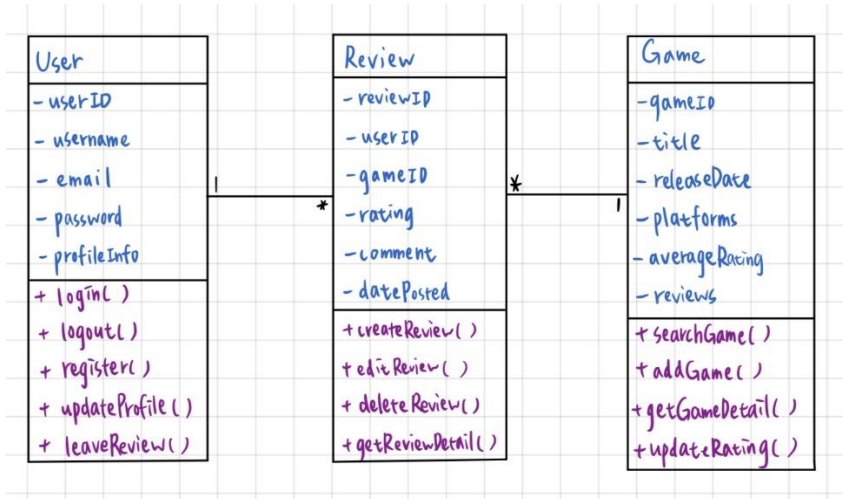
1. User signup
2. User login
3. Video game search
4. Filtered search
5. View user information
6. View game information
7. AI recommended games

## ***3. Architecture***

Familiarize yourself with Unified Modeling Language (UML).

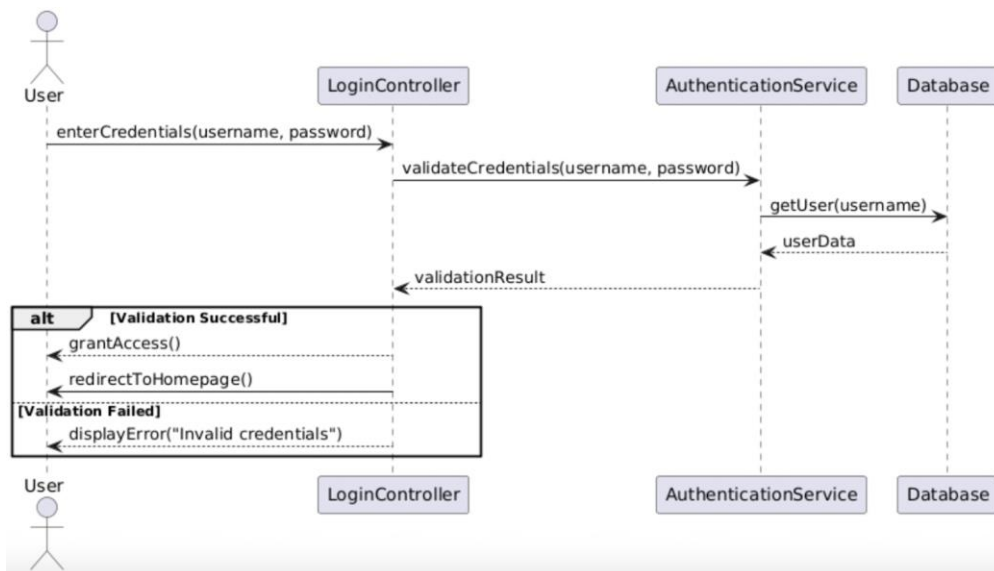
To implement your committed features, please provide

- High-level UML class diagrams for implementation classes of core functionality, Focus on main high-level classes only (one or at most two levels deep). This must reflect an OO approach to implementing your web site. For UML, you could find many references including <http://edn.embarcadero.com/article/31863>.

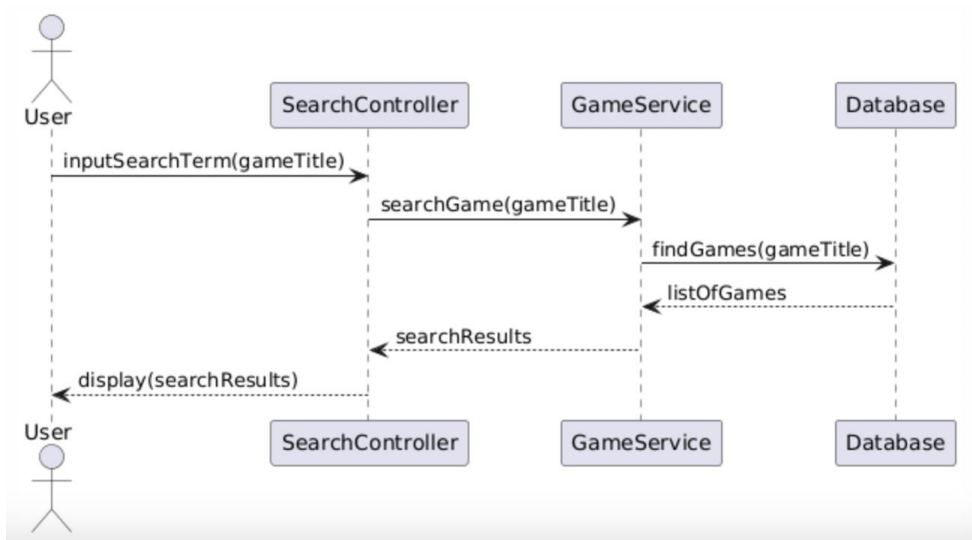


- High-level sequence diagrams: for ~5~6 functional requirements, please develop UML sequence diagram.

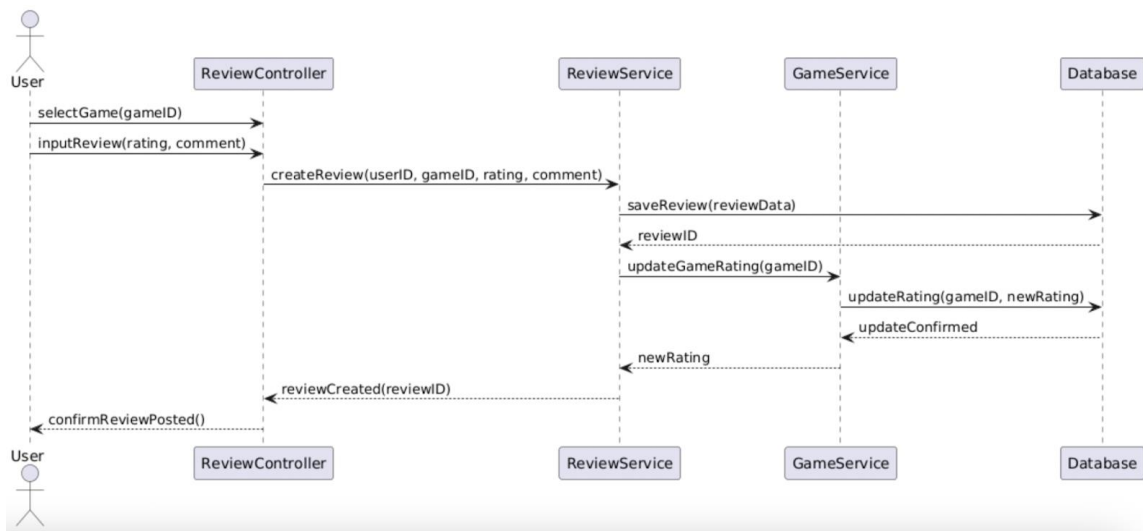
### High-level sequence diagrams: User Login Sequence



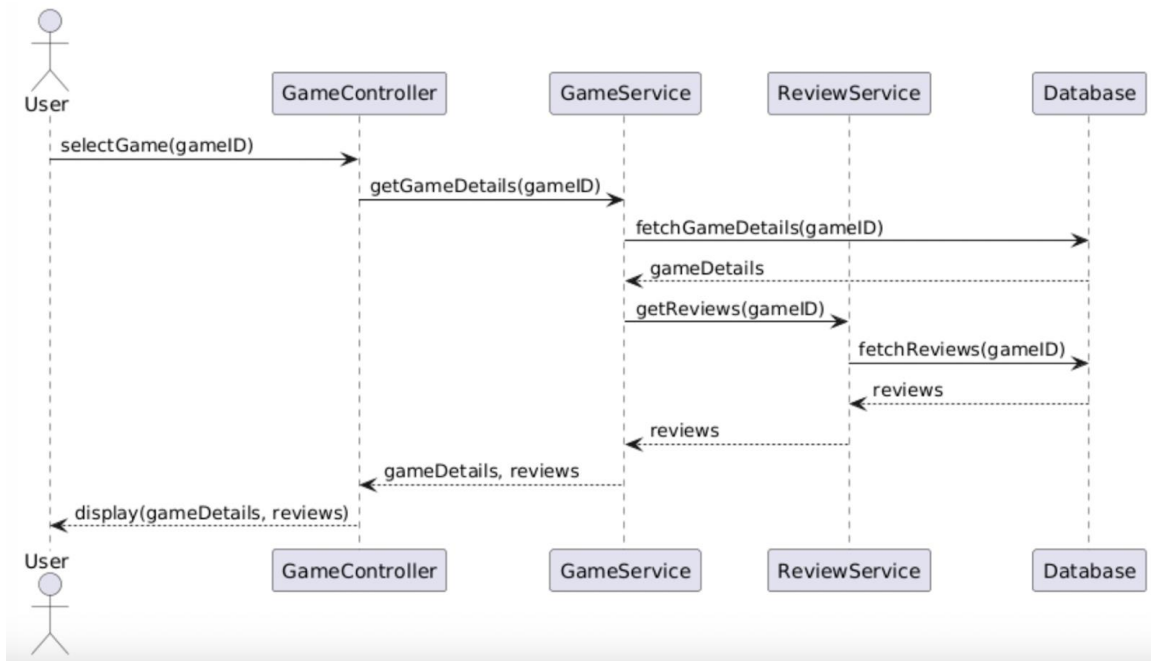
### Search Game Sequence



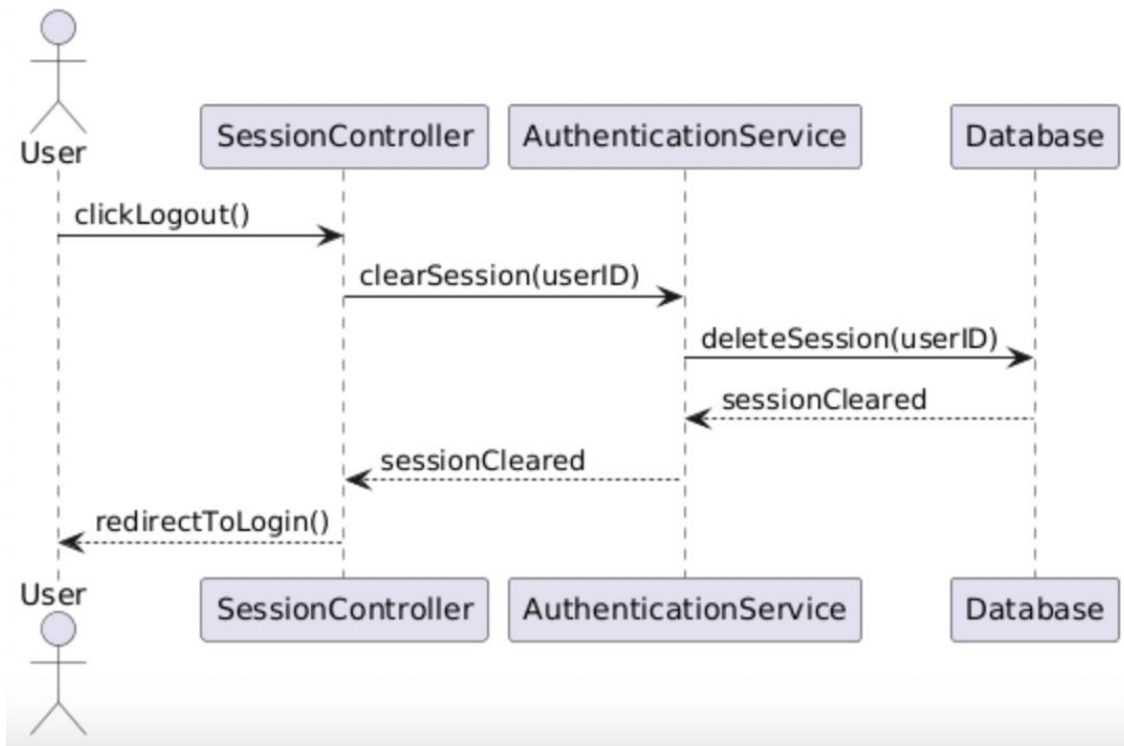
## Leave a Review Sequence



## View Game Details Sequence



### User Logout Sequence



#### **4. *Project status – write down the items before the demo and verbally explain it during the meeting if time is allowed***

- Risks: all actual (not hypothetical) risks (schedule, team work, technical, skills etc.) should be identified and either resolved or plans should be made to resolve them.

##### **1. Skills Risks and Mitigation Plan**

Risk: While the team has a good foundation in the required technologies (e.g., React, Node.js, MySQL), there is still a risk that members may struggle with new or unfamiliar technologies, such as the deployment process to AWS EC2 or advanced search functionalities.

Mitigation Plan: To address this, we have created a study plan that includes sharing tutorials, documentation, and online courses. Team members are encouraged to ask questions, explain their logic during meetings, and review each other's code to ensure that everyone is learning and improving. Additionally, tasks have been assigned to those most familiar with certain technologies to help bridge any gaps.

##### **2. Schedule Risks**

Risk: The primary risk here is if unexpected delays occur (e.g., bugs, feature misalignment) that could hinder progress on key milestones, impacting the final project deadline.

Mitigation Plan: Our team is using Trello for task management, where detailed tasks and deadlines are assigned to each team member. The tool is updated regularly, and if any changes occur, such as task reprioritization or delays, they are communicated transparently through the platform. We conduct regular check-ins and adjust the schedule as needed to ensure transparency and maintain momentum.

##### **3. Teamwork Risks**

Risk: While the team has generally been active, a potential risk lies in differing work paces and participation levels, which could cause bottlenecks in certain areas.

Mitigation Plan: Team members have been consistently attending meetings and adhering to their assigned tasks. In case of any delays or uneven pace, we plan to mitigate this by redistributing tasks when necessary and checking in more frequently with individuals who may need extra support. A clear communication channel (Slack) ensures that if anyone is stuck, they can quickly get help and stay on track.



**Appendix II– Rubrics and checklist for Part 2 Milestone 3 review: SW Review (to be done off-line by TA after Part 1 review)**

**Section: 01      Team: 02      Date:**

**Instructor/TA to Check and comment below:**

- Git/Github organization (e.g. organization of branches)
  - To setup Dev branch and Feature branches are strongly recommended.
  - Grading check point : Dev and feature branches are properly setup and used.
- Git/Gith, git hub usage: code review practices (to see if the review comments are proper and enough)
  - Grading check points :
    - how many code reviews are being done since M3 announcement
    - how meaningful code reviews are. (ex. “look good” does not have much meaning)
- Frameworks (back end front end) deployed correctly