

Release Date: 07/08/2025

General Instructions:

1. This assignment carries **40% out of the final marks.**
 - Continuous Assessments – 20%
 - Final Evaluation - 20%
2. This assignment must be done in groups of 4 members. Students must use the published grouping used for Y2S2 modules.
3. Late submissions will be penalized.
4. The Tasks deadlines and submission links will be published on the courseweb.

Each group should do the following.

1. You will implement the project identified in the HCI module in the Software Engineering module, following SE best practices, and proper project management processes.
2. Group leaders should join this GitHub Assignment and create the group with your group Number (example: "Group 1") <https://classroom.github.com/a/PNXcjcR>
 - a. The other members can then join your respective groups using the link.
3. Each group should maintain a JIRA project for their project.
 - a. Add all the members and assign responsibilities.
 - b. Add myself(Jeewaka.p@sliit.lk) and Mr. Chandana (students.chandana@gmail.com) to the projects.
 - c. Maintain proper epics, stories, and tasks.
4. Each member's scope
 - a. Every member must have an equal scope, which **covers all the aspects of development.**
 - b. This should include both Front-end and back-end.
 - c. **No member** should focus only on one aspect of development. (that is, Front-end or back-end)
5. **Implement the solution using React, Java Spring Boot as a Spring Web Application**
 - a. Develop the UI for the front-end Web application. It must be a responsive web application.
 - b. Backend (with Spring boot)
 - c. You are free to decide the database for the project based on the requirements

- i. SQL
 - ii. No-SQL
 - d. Each member should focus on applying OO Concepts and best practices, including **maintaining a GitHub repository and product backlog and following SOLID principles.**
 - e. A basic tutorial can be found here [Tutorial: Create your first Spring application | IntelliJ IDEA Documentation \(jetbrains.com\)](#)
6. The Application must be deployed and should be in working condition.
- a. You can deploy in any cloud service. (You have free credits from AWS, GCS, Azure)
 - b. Alternatively, other free services could be used.

Submission Requirements

- Students must use the GitHub Teams Repository assigned to them for coding. Individual contributions will be assessed based on the GitHub commits and activities.

Date	Activity	Marks
13th August	Project Details Submission	No Marks
18th August	Sprint Backlog Presentation	4%
1st September	Scrum Ceremony / Sprint Review 1	4%
15th September	Scrum Ceremony / Sprint Review 2	4%
29th September	Scrum Ceremony / Sprint Review 3	4%
13th October	Scrum Ceremony / Sprint Review 4	4%
20th October	Final Evaluation	20%

Grading Criteria
1. Sprint backlog Presentation

Weight in course: 4% of final grade

Evaluation: Group mark (all team members receive the same score)

Criterion	Weight	Poor(<25%)	Below Avg(25–45%)	Average(45–60%)	Good(60–75%)	Excellent(≥75%)	Group Mark
1. Sprint Goal & Scope	20%	- No clear goal or scope - No alignment with backlog tasks	- Vague or incomplete goal - Little explanation of scope or backlog alignment	- Goal is mentioned but incomplete - Scope partially addressed, missing key details	- Clear sprint goal - Scope mostly aligns with backlog items - Minor gaps in clarity	- Well-defined, focused goal - Scope is fully aligned and justified with backlog items - Thorough rationale provided	
2. Backlog & Task Readiness	20%	- Backlog very disorganized - No clear tasks or acceptance criteria	- Few tasks identified, many unclear - Minimal acceptance criteria or prioritization	- Some tasks defined, partial acceptance criteria - Basic prioritization but incomplete	- Well-defined tasks, each with acceptance criteria - Tasks logically prioritized in backlog	- Comprehensive task breakdown with clear acceptance criteria - Properly prioritized backlog - Thorough and consistent detail	
3. Timeline & Capacity Planning	20%	- No timeline or capacity discussed - No resource planning	- Rough or incomplete timeline - Poor capacity estimates, not well justified	- Basic timeline and capacity - Some attempt at resource allocation	- Realistic timeline reflecting team availability - Estimates generally accurate, minor adjustments	- Thorough timeline with clear milestones - Accurately reflects team capacity - Well-justified estimates and scheduling	
4. Risk Assessment & Mitigation	20%	- No mention of risks - No mitigation strategy	- Identifies only trivial risks - No clear or feasible mitigation steps	- Some key risks identified - Basic mitigation ideas	- Most major risks identified - Reasonable mitigation/contingency plans provided	- Proactive, in-depth risk analysis - Clear, robust mitigation strategies for each major risk	
5. Communication & Clarity	20%	- Disorganized or unclear presentation - Lacks logical flow, minimal explanation	- Some structure but hard to follow - Incomplete or unclear in major sections	- Generally coherent but with gaps - Uses minimal supporting materials	- Clear, well-structured, covers all necessary details - Uses some visuals/examples effectively	- Extremely coherent and engaging - Well-structured, polished materials - Excellent pacing and audience engagement	

2. Scrum Ceremony / Sprint Review

Weight in the course: 4% of final grade (per review) X 4 – 16% Total

Criterion	Weight	Poor(<25%)	Below Avg(25–45%)	Average(45–60%)	Good(60–75%)	Excellent(≥75%)	Student Marks
1. Demonstration of Completed Work	25%	- Little/no working features - Demonstration absent or unclear	- Very limited demonstration - Major functionality missing or non-functional	- Partial demonstration - Some functionality works, but incomplete	- Functional demo of most planned items - Minimal gaps in functionality	- Thorough demo of all planned items - Everything is functional and well-explained - Demonstration addresses acceptance criteria fully	
2. Team Collaboration & Communication	20%	- Very poor collaboration - One or two people dominate, others silent	- Sporadic contributions by some members - Communication lacks coherence	- Adequate overall collaboration - Some uneven participation	- Active participation from most members - Clear communication and role distribution	- Highly collaborative - Balanced, cohesive communication among all members - Roles, contributions, and handoffs are very clear	
3. Feedback Gathering & Action Items	25%	- No feedback discussion - No plan for improvements	- Minimal feedback acknowledged - Action items not well-defined	- Basic feedback gathered, some action items but lacking detail	- Feedback is discussed - Clear actions identified to improve next sprint	- Proactive, structured feedback session - Action items comprehensively documented - Integrates feedback into backlog	
4. Reflection on Sprint Process (Group)	30%	- No retrospective or reflection - Ignores what worked/didn't work	- Minimal reflection, lacks depth - Few suggestions for improvement	- Moderate reflection - Some mention of successes and failures	- Good retrospective - Identifies specific improvements for next sprint	- Insightful, thorough reflection - Well-defined, actionable strategies for continuous improvement - Difficulties Faced and Remedial Actions	

3. Final Evaluation(VIVA)

Weight in course: 20% of final grade

Criterion	Weight	Poor(<25%)	Below Avg(25–45%)	Average(45–60%)	Good(60–75%)	Excellent(≥75%)	Student Marks
1. Object-Oriented Concepts (OOC)(Individual)	20%	<ul style="list-style-type: none"> - Code is largely procedural or disorganized - Lacks fundamental OOP constructs (classes, methods, encapsulation) 	<ul style="list-style-type: none"> - Minimal OOP usage, classes exist but poorly structured - Frequent misuse of OOP features 	<ul style="list-style-type: none"> - Basic OOP structure - Some encapsulation/inheritance but with notable gaps or inconsistencies 	<ul style="list-style-type: none"> - Mostly strong OOP usage - Proper use of classes, inheritance, and encapsulation, with minor issues 	<ul style="list-style-type: none"> - Advanced OOP design - Thorough encapsulation, correct inheritance/polymorphism - Possibly includes relevant design patterns effectively 	
2. SOLID Principles (Individual)	20%	<ul style="list-style-type: none"> - Code shows “spaghetti” structure - Multiple large classes do everything, repeated violations of SOLID 	<ul style="list-style-type: none"> - Some attempt to separate concerns but multiple major violations remain - Minimal abstraction or modular design 	<ul style="list-style-type: none"> - Moderately applied SOLID - Partial success in single responsibility, open/closed, etc. but improvements needed 	<ul style="list-style-type: none"> - Generally SOLID-compliant - Code is modular, maintainable, and extendable - Only minor or isolated violations 	<ul style="list-style-type: none"> - Excellent adherence to SOLID - Highly maintainable, flexible, and extensible code - Classes and interfaces carefully designed 	
3. Presentation Skills(Individual)	20%	<ul style="list-style-type: none"> - Presentation is unstructured, unclear, or incomplete - Little to no visual aids, minimal explanation 	<ul style="list-style-type: none"> - Some structure but lacks clarity in major areas - Information is missing or poorly articulated 	<ul style="list-style-type: none"> - Moderately clear, some gaps in organization or detail - Uses a few visuals but could be more effective 	<ul style="list-style-type: none"> - Well-structured and clear - Good flow, explanations, and appropriate supporting materials 	<ul style="list-style-type: none"> - Polished, professional, and highly engaging - Smooth narrative, strong supporting visuals or demos - Excellent Q&A handling 	
4. Final Product & Deployment(Group)	20%	<ul style="list-style-type: none"> - Product is incomplete, non-functional, or extremely buggy - No deployment strategy or instructions 	<ul style="list-style-type: none"> - Minimal functionality or partially working product - Manual deployment with major issues, unclear steps 	<ul style="list-style-type: none"> - Core features function but some are incomplete - Basic deployment approach, possibly manual or poorly documented 	<ul style="list-style-type: none"> - All major features work as intended - Deployment is stable and reasonably documented (setup steps, config, etc.) 	<ul style="list-style-type: none"> - Product is polished, stable, and user-friendly - Deployment is well-documented and may be automated (CI/CD) - Clear instructions for end users 	

Group Assignment
SE2072 – Software Engineering Analysis
Semester 2, 2025

5. Git Repository Usage(individual)	20%	<ul style="list-style-type: none"> - Rare commits, no meaningful commit messages - No branching or collaboration strategy 	<ul style="list-style-type: none"> - Some commits but poorly messaged - Little evidence of merges or pull requests - Minimal branching 	<ul style="list-style-type: none"> - Basic commit practice with moderate clarity - Some branching but incomplete merges or code reviews 	<ul style="list-style-type: none"> - Good branching strategy - Consistent commit messages - Merges or pull requests used effectively 	<ul style="list-style-type: none"> - Exemplary repo management - Frequent, meaningful commits - Organized branching and merges, thorough PR reviews, clear commit history 	
--	-----	---	---	---	---	--	--

Important

- Please note all Group marks will be weighted based on the peer feedback from the group members.
 - IndividualScore = Sum of all **individual**-graded components.
 - Group Score = Sum of all **group**-graded components.
 - **p = Peer Feedback Factor**, which adjusts each individual's share of the group score (based on peer evaluations).
 - **Final Overall Mark**=IndividualScore + (p × GroupScore)
- Please inform us of any conflicts or issues faced as soon as possible. Do not bring the concerns to the last week of the semester.