Group Design Project
Work Plan and Deliverables

Week 1 – Familiarization (Mar 19 – Mar 23)

The objective for this week is to become familiar enough with the project to describe:

- architectural and structural features,
- potential risks,
- the key features and other important content from the existing backlog,
- the key scenarios currently supported by the project, and
- the core domain elements.

To support and document this investigation you will need to:

- Download and build the project.
- Install a code coverage tool,
- Run the test suite to capture coverage statistics
- Install a static analysis tool and use it to identify potential trouble spots in the code,
- Create a domain model that includes the top 10 – 20 most important objects,
- Create a high-level use case diagram, supported by
- Write detail level scenarios for the top 10 to 20 scenarios,
- Write a short document that identifies 3 to 5 features you might design (or redesign).

Deliverables (Due Mar 10 a.m. Mar 26)

Artifact	**Individual**	**Group**
Submit via Canvas:		
• A draft top-level structural diagram, consisting of at least one of the following: a package diagram, a deployment diagram, or a component diagram. If your project has separately deployed components, you must include a deployment diagram and one of the other two.	2 person partnership #1	
• A written report of what you consider the most significant problems uncovered by the static analysis tool. Identify the smell, the code involved, and why you think it is particularly serious.	2 person partnership #2	

• A use case diagram that represents the scenarios in the next item	x
• Written, detail level scenarios for the top 10 to 20 scenarios. Be sure to settle as a group on a template. Your template should include a field that identifies the scenario's author.	Each team member should write one fourth of the submitted scenarios
• A short document that describes three to five features your team might design and what you think they differ in difficulty, risk, and value.	x
• A domain model showing the domain elements most important to satisfying the scenarios written above.	x
• A tabular summary of these code metrics: count of non-comment lines of code, number of classes, number of packages, number of test classes, number of individual test cases, cyclomatic complexity (for highest scoring class).	x
Bring (as part of a working build) to your coaching session:	
The automated test results. Be prepared to show them run and discuss the results.	2 person partnership #1
A summary (keep a notebook) of any issues you encountered and questions you might have.	x
The static analysis results (ready to demonstrate a repeat run if asked). Detailed requirements to follow.	2 person partnership #2
A proposed set of tasks for week 2.	x
Key outcomes for coaching session	
• Selection of features to design/redesign. • Feedback for improving the documents. • Individual and partnership commitments for next week's tasks	

Week 2 – Impact Analysis (Mar 26 – Mar 30)

The objective for this week is to deeply investigate the selected project feature(s), propose a draft design, and do a first cut impact analysis. We are still performing analysis, but with one foot just across the line into design.

Deliverables, due April 2 at 10 a.m.

Artifact	**Individual**	**Group**
Submit via Canvas:		
• Detailed Scenarios for feature candidates	at least one from each individual. Competing proposals are encouraged.	
• Draft domain model showing revisions (i.e., new domain elements and domain elements that are expected to change) to support the new/revised scenarios.		x Alternatives are encouraged.
• An analysis of the quality of existing code associated with the affected domain elements	partnership #1	
• An analysis of existing tests that might be affected by the affected domain elements	partnership #2	
• A description of refactorings (design adjustments) that you think will ease the implementation or testing of the new/revised feature.		x
• A summary of key risks, what potential cost they carry, and what you might do to mitigate the risk or the potential cost.		x
Bring to your coaching session:		
• Your current working build	each individual should be prepared to demonstrate a working build and the ability to execute existing tests.	
• Design and work plan alternatives you would like to discuss. You will need to estimate times for this plan as part of next week's work.		x
• A list of tasks you think are necessary during the next week		x
Key outcomes for coaching session		
• Resolution of all tool chain issues. • Feedback on the existing documents. • Final team commitment to a specific design/project scope. • Individual and partnership commitments for next week's tasks		

Week 3 –Detailed Feature Design (Apr 2 – Apr 6)

The objective for this week is to produce a fully detailed draft design for the new feature. At this point we have both feet in design. Many of this week's deliverables will be refinements of previously submitted diagrams.

Deliverables, due April 9 at 10 a.m.

Artifact	**Individual**	**Group**
Submit via Canvas:		
• A textual description of your design strategy: what adjustments you plan to make to the existing code and why, what patterns you are using and why, what design rules have guided your selection of and revision of classes, what choices (if any) were influenced by issues in existing design, code, or tests.	2 person partnership #1	
• Detailed, feature specific use case scenario(s), supporting domain model, class diagrams, and interaction diagrams to support central use cases (enough to test domain model).	2 person partnership #2	
• Detailed interface descriptions for key interfaces.		x
• Sequence drawings to detail all interactions between classes in your feature and between your feature and existing classes.	divide drawings evenly among individuals	
Bring to your coaching session:		
• CRC cards sufficient to test a key scenario in person.		x
• A list of tasks you think are necessary during the next week		x
Key outcomes for coaching session		
• Clarity on what remains to polish the design. • Individual and partnership commitments for next week's tasks		

Week 4 – Final Feature Design and Draft Project Plan Estimate (Apr 9 – Apr 13)

The goal this week is to revisit and carefully document both the details in the final design and the differences between the existing and resulting designs, and to collect information from these details that we can use to construct a rough cost estimate, in terms of time.

Deliverables, due April 16 at 10 a.m.

Artifact	**Individual**	**Group**
Submit via Canvas:		
• A detailed summary of work to be done and an estimate of effort based on that summary. Work to be done is to be measured by counting classes, methods, tests which will be broken, and changes in the existing code, including refactorings. Suggest a staffing plan for three different project durations.	2 person partnership #1	
• A draft description of the development environment (see the section in the final report). A draft list of resources.	2 person partnership #2	
• Review of all earlier documents and diagrams to identify opportunities for improvement. Revise as much as practical now, with an eye to the final form required next week.		x
• Diagrams that highlight deltas between old and new, in all relevant aspects.	divide evenly among individuals.	
Bring to your coaching session.		
• Any questions about what is expected in the final report. • Any questions remaining from earlier weeks. • Any questions about the individual component due next week.		
Key outcomes for coaching session		
• Clarity on what remains to polish the design. • Individual and partnership commitments for next week's tasks		

Week 5 – Final Report (Apr 16 – Apr 20)

The objective for this week is to edit, update, and organize your work so that you can assemble it into a complete design proposal package. This is your chance to put your best foot forward.

Deliverables, due by April 20, 11:59 p.m. ***Note Friday due date!***

Artifact	Individual	Group
Submit via Canvas:		
The group's final report. This early portions of this report should be written as if you were describing the work to an executive committee who was charged with deciding whether to make the described change or not. The later sections are meant to be used by developers if the project receives a 'go'. The report should be organized thus: • **Introduction.** This section should give an introduction to the product associated with the project and to the feature(s) you have designed. This section is intended to help someone not familiar with the project understand what the product does, how it is typically used, why it has value to the users, what platforms are supported, and what your feature will contribute. • **Technical overview and context.** This section will present a top level technical overview of the product (as is). This section should include: use-case diagrams and textual scenarios, domain drawings and glossary/dictionary, appropriate top-level structural drawings, deployment drawings, and architectural descriptions/diagrams, key interfaces, and major patterns employed in the design. • **Development environment.** This section describes the development environment, including details of the assumed/compatible tool-chain, external dependencies (e.g., jar files, code generators), test framework, etc. • **Code Characterization**. Code metrics, test metrics, static analysis results, and your observations on code and design quality go here. Comment, as appropriate on unique coding conventions, special expertise or prior developer experience that may be important. • **Detailed Design.** This section explains the design you have chosen for your feature (give an abstract description first) and details the changes you propose in the existing design to support your design. This section should include before and		x

after domain drawings, class drawings, interaction and other behavior drawings, and (if appropriate) structural and package drawings. Required change points should be highlighted in all these drawings. Major interfaces and data structures should be fully detailed. • **Test Impact.** This section should attempt to quantify how much impact you think your design will have on existing tests. • **Level of Effort Estimates.** Here you present your estimate of the cost to implement the change example project durations. Your cost estimate should detail all your assumptions: i.e., identify the number of new classes and methods, the number of changes in existing production code (including refactorings) and the number of existing tests you expect will need rework. • **Resources.** Here you should give a list of web-sites, books, and other resources that you found helpful and that those who attempt to implement your design or do other work on this product might also find helpful.		
Your self-evaluation. This should address: • What was most and least valuable to you in this experience. • What was your most significant contribution(s) to the project. • What contribution(s) from others did you most appreciate. • What part of this work did you find most challenging (not most distasteful, but hardest to master.) • What do you think is the strongest aspect of the resulting feature design. • What kind of future change in the product do you think would be most likely to force your feature design to be significantly reworked. • What we should do differently to improve this exercise in future semesters. • What we should be careful to preserve/continue if we do this exercise again. • What you would do differently if you could do this project over.	x	