ASE 285 Team Project Document

- Software engineering is about rules and tools for building software products in a team. In this team project, students (a) learn how to build a REST web application and (b) contribute by adding new features using the Agile process.
- Students **should read** the `SE_project.pdf` to understand detailed ideas about the team project.

Project Team Players

1. Students should set up a team (of three), and select a team leader.
   - When we cannot make a team of three, we can make a team of four, but this is an exceptional case.
2. Team leaders are responsible for (a) communication with the professor, (b) managing schedule/team members, (c) making integration, regression, and acceptance tests, (d) leading sprint and scrum meetings, and (e) making documents (architectural diagrams and final documents).
3. Team members are responsible for making at least two non-trivial APIs and their unit tests and documents (manuals).
4. The professor and other team members become stakeholders.
   - Each team has an obligation to share the project progress (burndown report and any project-related issues) with stakeholders.
   - Each team leader has an obligation to upload/share the progress report on Canvas before

Timelines (deadlines and milestones)

Use the schedule word file for a specific deadline and milestones.

Prototype

1. We discuss the importance of prototyping to (a) know what we didn't know and (b) be on the same page with clients, but we don't make prototypes because we start the project with existing todoapps.
2. We have already finished the prototype to understand what we need to do for sprints 1 and 2.

3. The professor will make a project page for each team by the end of the prototype.
4. Before sprint 1, each team member decides what features to implement and shares the information using Canvas (HW4).

Sprint 1

1. We have two weeks for sprint 1.
2. We will have a short sprint review meeting to check and discuss the progress and results of sprint 1.

Sprint 2

1. We have two weeks for sprint 2.
2. Each team member makes two APIs with tests added to the existing todo app.
3. At the end of sprint 2 (Week 16), each team demonstrates its work and submits deliverables.

Sprint Goals

1st Sprint

A team leader should assign each activity to one or multiple team members.

- Goal 1 (Fix the bugs)
  - We make the id based on the document count, but when we delete a document and then add another document, to make the error. Fix this bug. For example, when we have id 1,2,3, and if we delete id 2. The app will assign a new id 3 to a new document to cause an error.
  - Using methodOverride to implement the edit POST API is unnecessarily complicated; revise the app by making the update API instead.
  - There is no error processing in the code; refactor the code to check if an error happens and displays correct/useful error messages.
- Goal 2 (Add and merge features)
  - For the list API, display the ID information.
  - Add at least one more item in the Navigation Bar.
  - Revise todo app 1 using the todoapp 2 features: Use Mongoose (for MVC architecture) and UIs (icon characters or CSS).

2nd Sprint

- Goal 1
  - Each team member implements features, and delivers the code, tests (unit/integration), and documents.

  - Each team leader makes tests - integration, regression, and acceptance.
- Goal 2
  - Each team member makes the manual part, and the team leader assembles the parts to make one user manual.
  - Each team leader makes the architecture of their application.

Example features/APIs (requirements) to implement

Students can implement any features, but these are some of the features that students can implement as the 2nd sprint goals. Students can (1) use them as a reference to come up with any features, (2) implement them, or (3) come up with any new features.

- Registration of users & creation of profiles using MongoDB.
  - API: register, unregister
  - Only registered users can access the app
  - After the registration, the registered (or unregistered) user information (profile) is displayed on TodoApp.
- Uploading and downloading a file
  - API: upload, download ...
  - Uploading a file into MongoDB or any cloud storage, download the files from the storage...
- Transform the todo app database results into a different format and download it
  - API: topdf, totxt ...
  - Transform any todo app database information into a single pdf or txt file which can be downloaded to users' computers.
  - After the transformation and download, the information about the files can be displayed on TodoApp.
- Tagging and/or indexing wiki pages
  - API: tag, index
  - Add a tag (or tags) to any data.
  - After tagging, we can display only the information that has proper tags.
- Search and advanced search
  - API: Regex search, adv-search
  - Regex search finds pages using patterns; advsearch finds pages based on "keyword", "date", or any criterion you choose to implement.
  - After the search, the matched information is displayed.
- Students can add any features with proper API names

Grading (250 points)

Use the grade rubric files for detailed evaluations.

Project Progress (50 points individual assessment)

- Team leaders will earn 100% when they make (a) all the backlogs, (b) share a progress report on Canvas, and (c) manage the project progress.
- Team members will earn 100% when they (a) finish (burndown) all the requirements/tasks with full test coverage and (b) send the report to the leader before the deadline.

Peer Review (50 points individual assessment)

- Use files in `references/peer-review/`.
  - Team players (leaders and members) upload the peer-review reports on Canvas before the deadline.
  - Team leader evaluates team members using the "references/peer review/team-leader-peer-review.docx."
  - Team members evaluate the team leader using "references/peer review/team-member-peer-review.docx."

Sprint 1 problem solving (team): 50 points

- Each team delivers the updated todoapp that solves all the issues and implements the required features.
  - The name of the students who solved the problem should be identified.

Project submission (team): 50 points

- Presentation (20 points): Each team will earn full points when they make a high-quality presentation file following the presentation rules.
- Documentation (30 points): Each team will earn full points when they deliver all the necessary documents.
  - Manuals and architecture diagram
  - Sprint meetings - retrospective and review

Project submission (individual): 50 points

- Team leaders will earn 100% when they submit all the tests and documents.
- Team members will earn 100% when they submit all the code/tests/documents for their features.