School of
Computing and
Information Systems

SMU
SINGAPORE MANAGEMENT
UNIVERSITY

Collaborative Software Development

Group Project

None Of Us Is As Smart As All Of Us

Ken Blanchard

Project Teams

- Students to form teams of 5-6 individuals
  - Work with the instructor to confirm team size
  - Finalize team formation before the end of 1st week
  - Notify the instructor if you can’t find enough team members
- Enroll to groups on eLearn
- No switching of teams allowed after the 1st week

Project Theme: *Climate Change*

Background: “Singapore Will Raise Climate Ambition to Achieve Net Zero Emissions By or Around Mid Century, and Revises Carbon Tax Levels from 2024”. Link: https://www.nccs.gov.sg/media/press-release/singapore-will-raise-climate-ambition

Example topics that students can develop an application for:

- Carbon neutrality solutions (measuring / tracking / analyzing)
- Promotion / awareness / education platforms
- Business / consumer oriented platforms that help drive the nation towards net zero emissions
- Sustainable investing, e.g., https://www.credit-suisse.com/sg/en/private-banking/invest-with-impact.html
- E-waste management: understanding and minimizing e-waste as part of a household/organization's hardware inventory

Your final product **must** have all the following components:

- **1. Backend REST API: user management and authentication, business logic (refers to any algorithms or calculations required by your app), and persistent database.**
- **2. User interface: command line, web, etc.**
- **3. Public deployment on any cloud platform.**

You are free to explore any potential use cases and applications in this theme.

Project Execution

- Agile process
  - Team to decide on stories, sprints, backlog, tracking, etc.
  - Team to assign own roles as project owner, Scrum master, etc.
- Tools
  - Java, Spring Boot (recommended)
  - Git (mandatory): Bitbucket/GitHub
  - Project management and tracking: Jira (recommended)
- Public deployment of services
  - Team to choose deployment platforms (AWS, Azure, etc.)

- Expected performance (B+)
  - Working prototype (code and deployment)
  - Evidence of agile process implementation (document)

- For better project grades
  - Any improvements over the expected performance
    - Scalable architecture, e.g., microservices; enhanced security features; good user interface, novel and interesting features, etc.

Project Grading (35%)

- Grading components:
  - **The software itself; and the software process**
- 1st milestone: proposal
  - Week 4: (2%)
    - 1-page free-form document to explain 1) what to do, and 2) why the need?
    - Feedback will be provided based on the usability, novelty and feasibility of the idea.
- 2nd milestone: first demo (8%)
  - Week 9: document (limit of 4 pages) and demo
    - Software process and technical details up to this point.
    - Source code (at least 1 sprint completed).
    - 5-min demo during class time.

Note that the page limit does not apply to the table of content and cover page for the documents. All documents to be submitted in Word or PDF format.

Project Grading (35%)

- Final milestone (20%)
  - Demo (15mins): week 13, in class
    - Functional app + any extra features
  - Submission: after the demo
    - 1 project document (limit of 10 pages: software process and technical details)
    - 1 project poster: 1 page with size A4
    - Source code (link to git repository in the doc – make sure it’s public)

- Peer-evaluation (5%)
  - Team member to rate the contribution of others
    - Proper justification needed for each rating

Note that the page limit does not apply to the table of content and cover page for the documents. All documents to be submitted in Word or PDF format.

Project Documents: Software Process

- Capture the agile process implemented
  - Screenshots needed, with important items highlighted and explained
- Capture details on the project backlog
  - User story organization
  - How the backlog evolved over the course of the project
- Capture details for each sprint
  - Sprint backlog
  - User story details
  - Sprint meeting/code review
  - Progress tracking
  - Sprint retrospective, lesson learnt, etc.
- Statistics obtained from Git repository for each team member: e.g., percentage of commits for everyone, etc. It's up to the team to include other measures of individual contribution.
- Details on any other things not listed here

Project Documents: Technical Details

- Software design: component based architecture, microservices, etc.
- API design and implementation
- User interface design
- Database design
- Security design
- Details on testing
- Details on continuous integration, deployment strategies, etc.
- Any other things beyond what listed here