Project Overview

Your course grades come almost entirely from the semester-long team project. The project will be written in Node.js and will use the Microsoft Bot Framework. Teams of 3-5 persons will be created by the instructor. Each team will be assigned a product owner by the instructor. Product owners may be a member of the instructional team, a person from industry, or a senior student.

Project Domains

Every group will be creating a conversation-bot that assists a student with some aspect of being a computer science student. In all cases the bots must go beyond the ability to answer simple keyword-based queries. Your group will select one of the following domains:

- A bot that identifies needs and then helps people connect to SOCS and UofG remote systems (i.e. linux.socs, wiki.socs, database servers, vpn, etc.)
- A bot that answers questions about C programming (i.e. explains error messages, helps with syntax questions, suggests programming constructs after asking about the problem being solved, etc.)
- A bot that answers questions about Java programming (i.e. explains error messages, helps with syntax questions, suggests programming constructs after asking about the problem being solved, etc.)
- A bot that helps the user decide who to contact given a conversation about the problem they are having and then helps them find the contact information for that person.
- A bot that provides program information for people in the BCOMP program (prerequisites, semesters courses are offered, content outline, required courses, area of application, etc.)
- A bot that uses an external wiki (like wiki.socs.uoguelph.ca) to assemble its own knowledge base and then answers questions from users about that knowledge.

Your product owner will flesh out more expectations for the capabilities of the bot within the domain you have selected.

User Interface Expectations

Your bot will be hosted on a linux server, provided by SOCS, that your team controls. You will provide a web front end to converse with your bot in addition to demonstrating the bot interacting with users in a slack or teams channel. You may add other user-interaction mechanisms if you wish, but the web front end and the slack/teams service are required.

Development Process: Scrum

We will follow the scrum software process for the entire project. The project will be divided into three sprints. For each sprint you will select user stories from the product backlog and implement and test those user stories. Your goal is to define user stories, in collaboration with your product owner, that can be completed in a single 3-week sprint. Your team will probably implement between 4 and 8 user stories per sprint. You likely won't complete all of the user stories in the product backlog by the end of the semester.

Each sprint must have a different scrum master. The product owner does not do any development but they own the product backlog.

Sprint meeting deadlines

Each of these meetings must include your product owner. Evidence of these meetings taking place must be included in your milestone summary. Try to meet these dates, especially for sprint review meetings.

- September 22: Last day to complete backlog generation
- September 29: Last day to complete Sprint 1 planning meeting
- October 15: Last day to complete Sprint 1 review meeting
- October 20: Last day to complete Sprint 2 planning meeting
- November 5: Last day to complete Sprint 2 review meeting
- November 10: Last day to complete Sprint 3 planning meeting
- November 26: Last day to complete Sprint 3 review meeting

Required Development Tools

- Microsoft Bot Framework must be used for the bot
- Node.js must be the main development language
- Project must be hosted in the provided repository on git.socs.uoguelph.ca
- Bot must be hosted on the provided linux server
- The project team must use the gitlab wiki, boards, and issue system to document and manage sprints

Grading Process

Your project is submitted as 3 milestones and a final presentation.

- Milestone 1 (25%): 8 a.m. October 16
- Milestone 2 (25%): 8 a.m. November 6
- Milestone 3 (25%): 8 a.m. November 27
- Final Presentation (10%): no later than 5 p.m. December 11 (this date is in the exam period)

Milestones

Milestones are simply a summary of the most recent sprint. Your milestone submission should be a single pdf file that provides evidence in the form of screenshots, links, and locations of the markable elements for that milestone. It is highly recommended that you use the list of markable elements as headings in your summary document. The submission may include extra explanation if desired. The list of markable elements for each milestone is listed later in this document.

If a group neglects to submit a milestone summary by the required due date, the instructional team will attempt to independently locate the markable elements in your git repository in order to grade the team's work. Some elements may get missed. **Your team is only eligible to request a milestone regrade if a summary document is submitted by at least one team member before the submission deadline.**

Milestone Grading

Milestones are graded out of 100. A maximum of 60 points will be awarded based on the process and completion of any 3 user stories (Sprint Backlog Items). The remaining 40 points are awarded for completing the optional items shown in the rubric. Your team may choose which optional items to select. If your team chooses no optional elements the maximum grade on the milestone is 60/100.

All optional elements should include evidence of the connection between the idea/concept and the concrete work you have done in your project. This could involve specific code examples (or counter-examples), wiki pages, design documents, or just a solid explanation. Graders need to know that your group understands the concept and how it connects to your project. Spelling and Grammar count. Paragraphs are not required. **If no summary document is submitted, no optional elements will be graded.**

Individual team members are required to assess the activities of each member of the team for each milestone. The assessment is done via the course website at the same time as the summary document is submitted. Any individual who does not submit an assessment of the team members will be penalized 10% of the milestone grade.

Milestone One (Grading Starts October 16)

Your team should select 3 of your Sprint 1 user stories to be evaluated for Milestone One. Selected user stories should not include housekeeping tasks like authentication or setting preferences.

Required Elements of Milestone	**Marks**
User Story #1 (no epics, proper format, acceptance criteria)	5
User Story #1 deliverable	5
User Story #1 repeatable testing	5
User Story #1 evidence that Scrum process was followed	5
User Story #2 (no epics, proper format, acceptance criteria)	5
User Story #2 deliverable	5
User Story #2 repeatable testing	5
User Story #2 evidence that Scrum process was followed	5
User Story #3 (no epics, proper format, acceptance criteria)	5
User Story #3 deliverable	5
User Story #3 repeatable testing	5
User Story #3 evidence that Scrum process was followed	5
Optional Elements: Choose any you wish up to a maximum of 40 possible points	
Agile Development: Clear description of the roles for each team member for this sprint. Summary/Minutes from all meetings.	20
Agile Development: Evidence of team processes including merge review, coding standards, use of a linter, build tools (CI is optional)	20
Scrum Processes: Evidence of extra attention to scrum process including well-developed scrum boards, a groomed backlog, use of issue tracking (including regular issue updates, and documentation in wiki on GitLab)	20
Code Smells: Discussion of how any 3 code smells have been avoided or refactored out. Discussion should include definitions and reasons why those code smells are likely/common to this type of project.	20

Milestone Two (Grading Starts November 6)

Your team should select 3 of your Sprint 2 user stories to be evaluated for Milestone Two. Selected user stories should not include housekeeping tasks like authentication or setting preferences.

Required Elements of Milestone	**Marks**
User Story #1 (no epics, proper format, acceptance criteria)	5
User Story #1 deliverable	5
User Story #1 repeatable testing	5
User Story #1 refactored	5
User Story #2 (no epics, proper format, acceptance criteria)	5
User Story #2 deliverable	5
User Story #2 repeatable testing	5
User Story #2 refactored	5
User Story #3 (no epics, proper format, acceptance criteria)	5
User Story #3 deliverable	5
User Story #3 repeatable testing	5
User Story #3 refactored	5
Optional Elements: Choose any you wish up to a maximum of 40 possible points	
Testing: Evidence of a full suite of unit tests with coverage reported and a test plan for integration testing.	20
DevOps: CI/CD employed for linter, unit tests. Merge rules require pipeline success. Single-issue branch/merge strategy in place for project	20
SOLID: Discussion and evidence of adherence project wide to at least 3 of the 5 SOLID principles. Discussion should include reasons why any of the 5 are not included.	20
Team Choice: Relate a course topic not previously submitted for marks to your project with evidence	20

Milestone Three (Grading Starts November 27)

Your team should select 3 of your Sprint 3 user stories to be evaluated for Milestone Three. Selected user stories should not include housekeeping tasks like authentication or setting preferences.

Required Elements of Milestone	**Marks**
User Story #1 (no epics, proper format, acceptance criteria)	5
User Story #1 deliverable	5
User Story #1 repeatable testing	5
User Story #1 refactored	5
User Story #2 (no epics, proper format, acceptance criteria)	5
User Story #2 deliverable	5
User Story #2 repeatable testing	5
User Story #2 refactored	5
User Story #3 (no epics, proper format, acceptance criteria)	5
User Story #3 deliverable	5
User Story #3 repeatable testing	5
User Story #3 refactored	5
Optional Elements: Choose any you wish up to a maximum of 40 possible points	
Refactoring: Identify code improvements between M1 and M3. Explain the benefits process for refactoring done in at least 3 sections of the code.	20
Architectures: Explain the architecture used for this project. Identify another possible architecture and explain how it could be accomplished.	20
Design Patterns: Identify and document the existence of any three design patterns in your codebase	20
Team Choice: Relate a course topic not previously submitted for marks to your project with evidence	20

Final Presentation (no later than Dec 11)

Your final presentation may be a virtual presentation via Zoom done in real time (and recorded for your peers to see if they can't attend) or it may be a pre-recorded video presentation. You may be asked to answer questions during the live presentation or answer written questions if you submit a pre-recorded presentation. Presentations will be marked out of 30 points. The maximum possible grade is 30 but you may include as many elements as you want and we'll just cap your grade when/if we reach 30. If your presentation is missing the required elements it gets a grade of 0. Grammar, diction and polish count.

Required	**Marks**
Statement of purpose of the bot	5
Demonstration of the bot in use via Slack	5
Demonstration of the web front end of the bot	10
Optional	**Possible Marks**
Summary of user stories completed	5
Future work (highlights from the remaining product backlog)	5
Most valuable skills developed (individual team members or consensus)	5
Knowledge gained/learned (individual team members or consensus)	5
Team member introduction and major contributions	5
Product owner introduction	5