

# **Computer Science Senior Project**

## **Prerequisites**

The courses **Programming Languages** and **Design and Analysis of Algorithms** are prerequisites for this course.

## **Catalog Description**

This is an advanced programming course in which students work on large-scale individual or team programming projects and make a formal presentation on their work. The course discusses program development, methodologies and strategies.

## **Professor**

Name: Dr. Baliga  
Phone: (856) 256-4500 ext. 3890  
Email: [baliga@rowan.edu](mailto:baliga@rowan.edu)  
Office hours: Thursday 12:30-3:30

## **Instructional Methods and Techniques**

### **Project Groups**

Students work in groups of 6-8 students. Groups should be formed as soon as possible. Each group will elect a group leader who will email the professor about the group's composition by the group formation deadline (refer to section *Important Dates and Deadlines* below). After this deadline, the composition of the groups is fixed for the rest of the semester. It is understood that all members of a group will contribute significantly towards the project. Each group will create an online repository (e.g., on GitHub, GitLab or Bitbucket) for all project artifacts (documents, code, etc.) All group members will actively contribute to updating the repository on a regular basis throughout the semester. The professor must be provided read access to the repository. A group member's git commit history will be used to judge their contribution to the project.

### **Project Specification**

Groups are encouraged to choose their respective projects. Please ask the professor for help if you are unable to find a project. The project specification (spec, from here on) will be completed by the project spec deadline. The spec will comprise

- Title page containing
  - Project title
  - The names of the team members.
  - The name of the team leader. The team leader is the team's first point of contact for the professor.
  - The project's GitHub site
    - The project GitHub *must* contain at least one commit from each team member by the spec deadline. Note: Private repositories are acceptable; the instructor must be provided with read-only access.
  - The GitHub site for each team member
  - Your team's Slack (at slack.com) workspace. The instructor must be invited to join the workspace by the spec deadline.
- Project summary
- Project goals
- Product features
- Any limitations
- Any stretch goals

Essentially, the spec describes important capabilities of the program that you are setting out to create. While writing this document, feel free to research, refer to and include articles, URLs and anything else that may help clarify what you are setting out to achieve. The spec establishes a minimum requirement for the final project submission. Please meet with the instructor before the project spec deadline with draft versions of the spec for feedback. The instructor must approve any change(s) to the project spec after the spec deadline.

## Analysis and Design

There are two deliverables in this phase. Firstly, the team will create a design document for the project. Secondly, each team member will create an individual prototype which will be uploaded to their individual GitHub.

After careful analysis of the spec, each group will design a solution to the proposed problem. You **must** consider multiple approaches and their respective pros/cons for satisfying the spec's requirements. The approach to be used will be described in a design document, which is due by design document deadline. A design document for a mobile app should minimally contain/address:

- High level description of the design that includes the system architecture
- Problem solving approaches that were considered during the design phase. You are required to document the two most promising ones that were considered and rationale for selecting the winning approach.
- Mockups of the different screens in the app. Scans of (neatly) hand-drawn screens are acceptable.
- Navigation between app screens
- Backend information
  - Database schema: Describe the attributes for all tables in your database.
  - RESTful API: A list of all RESTful endpoints that will be implemented. For each endpoint, list the endpoint's functionality and it's input and output payloads.
- Functionality provided by each app screen. Explain which RESTful endpoints are accessed by the screen to provide the specified functionality.
- User authentication and data security issues
- Tech stack (platform [Android/iOS etc.], development tools [e.g. Android Studio, Flutter etc.], RESTful APIs [e.g. Google Maps for geolocation, etc.], backend [e.g. AWS EC2 Linux/Python Flask/MySQL etc.], programming language(s), etc.) that has been selected to implement the project
- For the mid-assessment,
  - list goals to be accomplished
  - list specific tasks assigned to individual group members.

Each team member will **individually** build a prototype that demonstrates their understanding of one or more components of the tech stack that is being adopted for the project. This prototype will be published to the individual's GitHub by the design document submission deadline. The prototypes built by team members will collectively demonstrate the team's ability to use all crucial technologies in the selected approach. Please meet with the instructor before the design document deadline for feedback on preliminary versions of the design document and to discuss ideas for your individual prototype.

## Implementation and Demo

The implementation/testing phase will start immediately after the design document is submitted. There are two sprints in this phase. At the end of the first sprint is the mid-assessment demo. The final project demo is at the end of the second sprint. *Your project must be fully functional at final demo time.*

## Classroom Presentation and Final Submission

Classroom presentations are scheduled on the day of the final exam. The final submission entails fixing any bugs that are observed during the demo, updating the project GitHub site with the latest project artifacts such as the classroom presentation deck, and, making any changes suggested by the instructor.

## Grading

There are no exams in this course. Successful completion of this course strongly hinges on accomplishing **all** the stages of the project **in a timely fashion** and, above all, the successful completion of **all** the goals listed in the spec. The deadlines listed for various project deliverables are as listed below. Failure to meet a deadline may result in the loss of the corresponding assigned grade. To pass this course, you **must** earn a passing grade in **each** of the nine components listed below.

1. Project specification document (spec) **(5%)**.
2. Design document **(15%)**
3. Individual Prototype **(10%)**
4. Mid-assessment **(10%)**
5. Final demo **(10%)**
6. Classroom presentation **(15%)**
7. Documentation **(10%)**
8. Individual contribution **(10%)**: Primarily assessed using project GitHub commit history and peer assessments
9. Overall project quality **(15%)**

The correspondence between percentage scores and letter grades will be as follows: 93 and above (**A**), 90 to 93- (**A-**), 86 to 90- (**B+**), 82 to 86- (**B**), 78 to 82- (**B-**), 74 to 78- (**C+**), 70 to 74- (**C**), 66 to 70- (**C-**), 62 to 66- (**D+**), 58 to 62- (**D**), 54 to 58- (**D-**), less than 54 (**F**).

The deadlines listed for project stages are firm. Please work in a timely manner. Failure to meet a deadline may result in the loss of the grade assigned to that stage.

## Important Dates and Deadlines

Group formation:	9/10
Project spec:	9/17
Design document:	10/8
Individual prototype review:	10/10 – 10/16
Mid-assessment demo:	10/29 – 10/31
Final demo:	12/3 – 12/5
Presentation:	Scheduled final exam day 12/12 – 12/18
Final submission:	12/18