CREDERA

Software Engineering II
Baylor University
Fall 2023

Table of Contents

1. Objective **2**

2. Expectations **3**
2.1 Students **3**
2.2 Baylor **3**
2.3 Credera **3**

3. Project Details **4**
3.1 Project Assessment **4**
3.1.1 Problem Statement 4
3.1.2 Milestones 4
3.1.3 Schedule 5
3.2 Office Hours **6**
3.3 Group Member Roles **6**
3.4 Product Backlog **7**

4. Technology Frameworks **8**
4.1 Technology Stack **8**
4.1.1 Spring 8
4.1.2 React. 8
4.1.3 REST 8
4.1.4 Google Cloud Platform 8
4.1.5 MySQL 8

5. Credera Mentors **9**

1. Objective

Credera seeks to interact with students and faculty to develop deeper relationships through a semester-long project in Software Engineering II. Students will cooperate in teams, with the professor and Credera mentors serving as guides to complete the project.

2. Expectations

2.1 Students

Students should expect to face a real-world situation where they will be given a problem and are tasked with developing a solution. They will need to understand how the problem is approached, propose a solution, and implement it. They will work in teams, just as commonly occurs on Credera projects, to solve the problem and present their findings, emulating actual work practices on Credera projects. This will not be an easy task. Students will be expected to research and find solutions to their team's issues. Mentors are there to assist, but not to solve challenges. Students will be expected to come to their office hour/scrum meetings prepared with specific questions and deliverables complete.

2.2 Baylor

Baylor will partner with Credera to manage the project to better prepare their students for jobs following graduation. Professors will work as a point of contact for Credera and the students on all project questions. Professor will work with Credera to ensure project meets all class expectations.

2.3 Credera

Credera will need four employees to participate in on-campus activities and regular virtual meetings with the students. Through weekly calls with the students, Credera employees will be expected to mentor in best practices and problem-solving solutions. Credera will be expected to work with professors to meet class expectations. Deliverables will be given to the students and mentors are expected to review and offer feedback to students.

3. Project Details

3.1 Project Assessment

3.1.1 Problem Statement

A new company, StudyBuddies, Inc., wants to create a custom experience for students and tutors alike to allow them to plan study sessions anywhere, on and off campus. You've been hired as a consultant to design and build a web application for this project that allows connecting with other users, creating and scheduling meetups, recommending locations to study at, and more. The client wants your team to create a user-friendly web app for users to register under the appropriate classification, input their class(es) and/or area(s) of study, and find other students or tutors with similar profiles. In addition, your team can enhance the application with other features at your discretion.

3.1.2 Milestones

This semester will be broken out into four milestones for technical implementation. Each team will be responsible for completing each milestone by the assigned week (see project schedule for dates). At the end of each milestone, each team will present their current implementation to the mentors and professor during scrum. A formal presentation is not needed during the first three milestone checkpoints, but an informal yet thorough walkthrough of current complete functionality will be expected.

Milestone 1:

- Have team repository set up
- Have pages scaffolded out with routing (can be empty, home page just says "Home", etc…)
- Call example Java endpoint from client (ask the mentors for more details on this in our meetings)

Demo:

- Demonstrate that all the pages exist by directly hitting the URLs
- Display the result of the test endpoint service call

Milestone 2:

- Have all the pages implemented – sign-in/registration, profile page, home page (at a minimum)
- In the service layer, be hitting a MySQL Database and be able to create/edit account, and add/edit classes/areas of study, add/edit "study partners", add/edit profile details, create/edit meetups, review tutors, etc.
- Have deployed the template project to GCP
- Have continuous integration setup (a Git commit triggers a build, which triggers a GCP deployment)
- In the front-end, the individual should be able to:
  - Register and log in
  - Select user type (student vs tutor) during registration
  - Add and edit classes/areas of study
  - Search for and connect with other users
  - Create and edit study meetups
- Add unit tests in Java layer and add a testing step to the continuous integration deployment pipeline (maintain and update tests moving forward)

Demo:

- Show the app running live on GCP
- Change the message on the home page and commit the change, then check if the continuous integration is working and the message changed on the live site
- Demonstrate creating a new account and logging in
- Demonstrate creating a new study meetup and providing details (date, time, location, subject, etc.)
- Show the deployment pipeline, with all tests passing

Milestone 3:

- In the service layer, be able to suggest other users to connect with, schedule, edit, and cancel meetups, send notifications, and submit ratings/reviews for tutors.
- Recommendations: A critical component of this project is a sophisticated system for connecting users based on shared interests. Typically, teams achieve this by allowing the individual to enter subjects they

wish to study, select whether they are searching for students and or tutors, and set preferences for study methods. Then, the system can recommend a list of other users and the individual can choose to connect with and schedule meetups with any number of them. You can take a different approach from the one above and even add components like a study playlist or location recommendations, but it needs to be able to create a study meetup effectively and also support a large user base.

- In the front-end, individuals should be able to:
    - Review recommended tutors
    - Connect with and join/schedule meetups with recommended tutors
    - Get meetup and connection notifications
    - Edit or cancel a meetup
    - Rate the performance of a tutor after a tutoring session
    - Update existing and add new unit tests in Java layer

Demo:

- Suggestions for students and tutors to connect with, related to study preferences and subject
- Reminders before meetup
- Rating a tutor after a tutoring session

Milestone 4 (due at time of Final Presentation):

- Security features (viewing other user's connections and meetups, only editing user-owned or shared meetups, unauthenticated user viewing the site, etc.)
    - Input validation
    - User authentication
        - For example, a validated user can edit only their own classes/areas of study, but can see others' in a read-only format; unauthenticated users receive a different view of the site, etc.
    - Password encryption
- Final UI/UX design
- Bonus features

Demo:

- Demonstrate security
- Demonstrate bonus features
- The UI should be complete
- Product and functionality should be in a complete state

Bonus Feature Examples:

- Socializing/gamifying the app
- Tutorial
- OAuth sign in through a social media site like Google/Facebook/Twitter
    - There is an approval process to set up some OAuth integrations, if you choose to do this be sure to start early
- Social media integrations
- Calendar integrations
- Mobile/out-of-app notifications
- Playlist generator/recommendations
- Location recommendations (e.g. places with public Wi-Fi, student discounts, etc.)
- Password reset
- Mobile application
- Promotions
- Use of the Alexa service
- Ensure your application is compliant with the OWASP Top 10

3.1.3 Schedule

The following is the schedule for this semester along with the due dates of milestones and weekly deliverables.

Date	Week	Credera Assignment
8/28	1	**Project Kick Off!** (3:30 – 4:45pm) • Class introduction • Project kick-off presentation Role assignments and requirements clarifications
9/4	2	• First scrum meeting • Continued requirements clarification for Milestone 1 • Milestone 1 development • Deliverable 0 (Project Setup) due
9/11	3	• Milestone 1 development • Deliverable 1 (ERD) due
9/18	4	**Milestone 1 Due** • Continued requirements clarification for Milestone 2 • Begin Milestone 2 development
9/25	5	• Milestone 2 development • Deliverable 2 (Account Creation & Login) due
10/2	6	• Milestone 2 development • Deliverable 3 (E2E Service Demo) due
10/9	7	**Milestone 2 Due** • Continued requirements clarification for Milestone 3 • Begin Milestone 3 development
10/16	8	• Milestone 3 development
10/23	9	• Milestone 3 development • Deliverable 4 (Recommendations) due
10/30	10	• Milestone 3 development • Deliverable 5 (Notification) due
11/6	11	**Milestone 3 Due** • Continued requirements clarification for Milestone 4 • Begin Milestone 4 development
11/13	12	• Milestone 4 development • Deliverable 6 (UI Hardening) due
11/20	13	• Thanksgiving break – no scrum
11/27	14	• Milestone 4 development
12/5	15	**Final Presentation (3:30-4:45pm) / Milestone 4 Due**

3.2 Office Hours

Office hours will occur weekly. Teams will have a 20-minute slot on a day determined by the students in the evening. At least two mentors will be present for every scrum, but an individual mentor will not be assigned to a specific team; rather all mentors will be available for all teams. Every team member is responsible for stating what they accomplished last week, what they plan to accomplish this week, and any roadblocks. The remainder of the time will be used on some of the following:

- Each team should have questions ready (i.e., requirements, Spring, APIs) to discuss during the meeting for further understanding
    - Feel free to send the mentors questions ahead of time so that we can better utilize scrum time
    - Any code needed for questions should be available in source control so that the mentors can look at it directly and run it locally
- Review weekly deliverable or discuss the upcoming deliverable
- Provide a brief demo of new functionality

Questions submitted asynchronously will be answered in a timely manner, but please note project mentors will most likely be confined to business hours, meaning questions may not be answered until 24 hours later. Please plan questions outside of office hours accordingly.

3.3 Group Member Roles

The minimum prescribed roles in each group are as follows:
Project Manager will be responsible for coordinating the activities of the group and for liaison with the professor/coach to resolve issues affecting the group from time to time. The project manager should be: professional and responsible, a good organizer, and an effective communicator. His/her functions include: 1) Coordinating group activities, 2) Overseeing and managing activities, 3) Allocating roles, activities and responsibilities, Coordinating and managing documentation activities (You may refer to What a team leader does/does not as well.)
Requirements Engineer leads the requirements effort
Design Engineer leads the design effort
Quality Assurance Engineer oversees test case design, validation of requirements, design, etc.
Project Librarian keeps all meeting logs and makes all design artifacts available for the team

The initial role assignment will be decided within the group and can change during the semester, if needed, after consulting with the professor/mentors. Each student is also required to summarize the activities they carried out and give the times spent on each. Regardless of group member role, each team member is expected to make contributions to the code-base.

3.4 Product Backlog

The product backlog includes a list of artifacts that must be completed this semester. The following list must be included in each team's final deliverable.

Product Backlog
Build activity diagrams - Initial use cases outlining, in general, core user interactions with the new feature
Build use case documents - Brief listing of requirements generated from use case diagrams
User interface mockups (if applicable) - UI mocks based on the use cases to help showcase how the feature will be implemented
Create requirements class diagrams - Class diagrams be built off the requirements & use case document. Students need to keep in mind that requirements class diagrams are meant to show how the domain is related, not how it is implemented, and the ER diagrams show the actual database tables; correlating them may take some work
Develop architecture diagram - An initial architecture diagram that outlines major elements of the feature and how they relate
Build sequence diagrams - Initial sequence diagrams showcasing major flows in design class diagram
Document project plan – The project plan outlines deliverables. The students should indicate how much of the feature they can complete and what parts of the feature will be accomplished by each deliverable in detail
Create test plan – The test plan details how features are tested in each milestone. Testing should include both functional and unit testing, and a milestone should be considered complete when the tests are satisfied
Document design patterns used – Documentation showcasing which patterns you have selected and how you are implementing
Create User Manual – Guide should detail operations and maintenance of your project
Final project documentation - Document organizing all prior deliverables
Final project source code
Final presentation - PowerPoint presentation or otherwise that demos your project, you should talk about the software development process and any risks/setbacks/difficulties encountered along the way

4. Technology Frameworks

4.1 Technology Stack

4.1.1 Spring

Spring (aka Spring Framework) is an open-source framework and inversion of control container for Java that was created to address the complexity of developing enterprise applications. Spring has layered architecture that allows developers to leverage certain components while not utilizing others that they may not care about. Spring's biggest features include:

- Transaction management – Spring allows for pluggable transaction managers to deal with transactions
- Inversion of control container – Spring allows for dependency injection, which helps with the configuration and management of Java objects
- Data access – Easy integration with Hibernate and JDBC
- Model-View-Controller (MVC) – Spring provides a framework for extending and customizing web applications
- Messaging – Spring can leverage existing technologies, such as Java Messaging Service (JMS) for sending messages
- Some of these technologies will be vital to your project, and Spring allows for easy integration with other useful technologies and frameworks.

4.1.2 React

React is a JavaScript-based open-source web application framework for the application's front-end that allows you to create dynamic views for your single-page application. It is a very widely used platform and is currently one of the most popular projects on GitHub.

A popular UI component/styling library called Material UI is included and pre-configured in the template project but is not required to be used.

4.1.3 REST

Representational state transfer (REST) is an architecture approach (not a tool or language) that consists of a coordinated set of constraints that applies to components, connectors, and data elements. The REST architecture has been applied to developing web services and is commonly used as an alternative to SOAP. REST allows for data to be quickly obtained via point-to-point communications with clients. For this project, REST will be used to obtain data from the service layer.

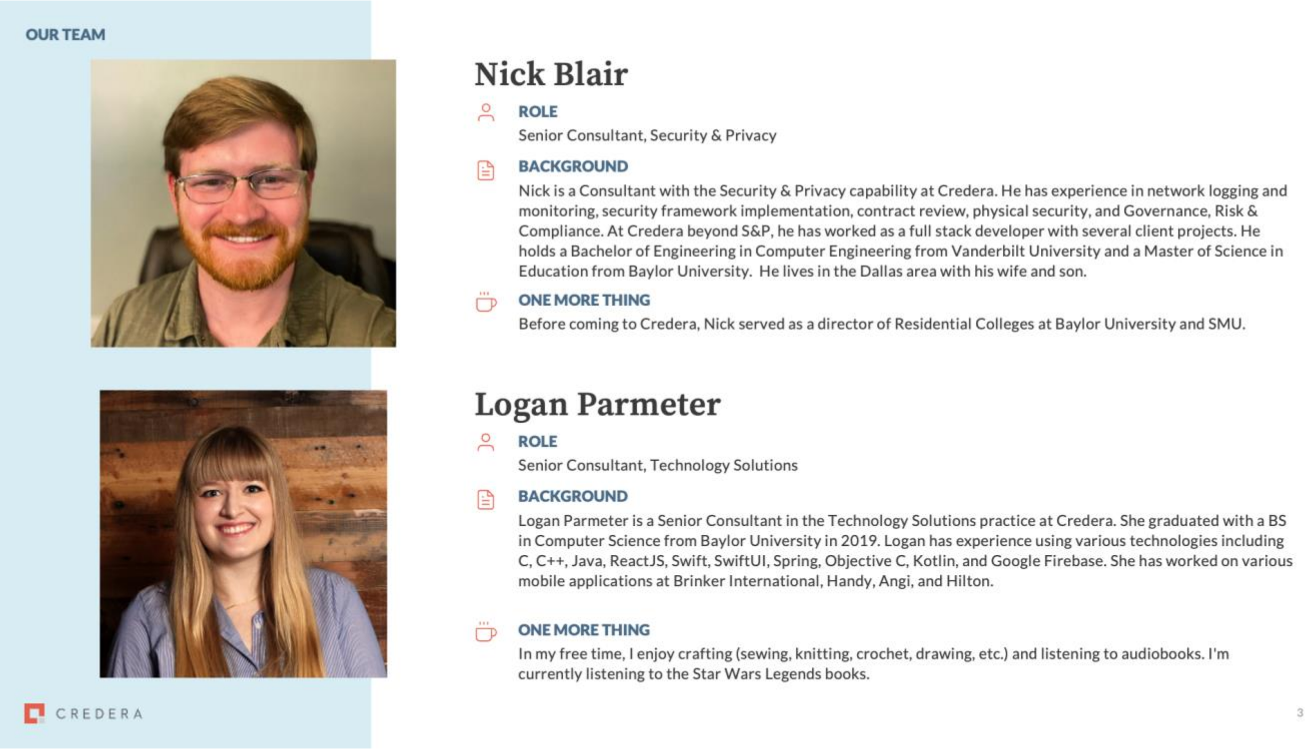
4.1.4 Google Cloud Platform (GCP)

GCP is Google's cloud platform. You will use GCP to host your application so that it will be publicly accessible on the web. You will be able to control how the application deploys, runs, and connects with your database and any other resources.

4.1.5 MySQL

MySQL is a popular open-source relational database management system. It is reliable and easy to use and provides official Docker images maintained by the MySQL team for Linux platforms. MySQL is a leading choice for SQL databases and is used in a wide variety of applications.

5. Credera Mentors



OUR TEAM

Nick Blair

ROLE

Senior Consultant, Security & Privacy

BACKGROUND

Nick is a Consultant with the Security & Privacy capability at Credera. He has experience in network logging and monitoring, security framework implementation, contract review, physical security, and Governance, Risk & Compliance. At Credera beyond S&P, he has worked as a full stack developer with several client projects. He holds a Bachelor of Engineering in Computer Engineering from Vanderbilt University and a Master of Science in Education from Baylor University. He lives in the Dallas area with his wife and son.

ONE MORE THING

Before coming to Credera, Nick served as a director of Residential Colleges at Baylor University and SMU.

Logan Parmeter

ROLE

Senior Consultant, Technology Solutions

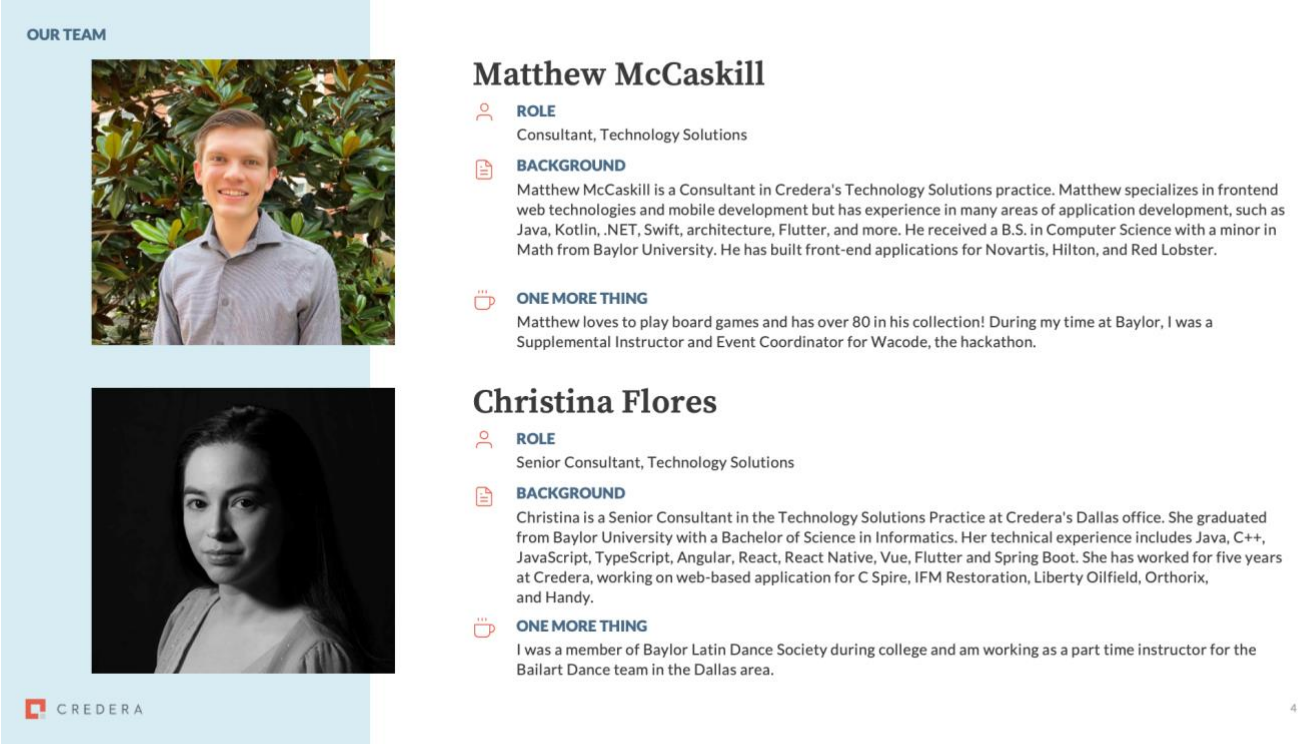
BACKGROUND

Logan Parmeter is a Senior Consultant in the Technology Solutions practice at Credera. She graduated with a BS in Computer Science from Baylor University in 2019. Logan has experience using various technologies including C, C++, Java, ReactJS, Swift, SwiftUI, Spring, Objective C, Kotlin, and Google Firebase. She has worked on various mobile applications at Brinker International, Handy, Angi, and Hilton.

ONE MORE THING

In my free time, I enjoy crafting (sewing, knitting, crochet, drawing, etc.) and listening to audiobooks. I'm currently listening to the Star Wars Legends books.

3



OUR TEAM

Matthew McCaskill

ROLE

Consultant, Technology Solutions

BACKGROUND

Matthew McCaskill is a Consultant in Credera's Technology Solutions practice. Matthew specializes in frontend web technologies and mobile development but has experience in many areas of application development, such as Java, Kotlin, .NET, Swift, architecture, Flutter, and more. He received a B.S. in Computer Science with a minor in Math from Baylor University. He has built front-end applications for Novartis, Hilton, and Red Lobster.

ONE MORE THING

Matthew loves to play board games and has over 80 in his collection! During my time at Baylor, I was a Supplemental Instructor and Event Coordinator for Wacode, the hackathon.

Christina Flores

ROLE

Senior Consultant, Technology Solutions

BACKGROUND

Christina is a Senior Consultant in the Technology Solutions Practice at Credera's Dallas office. She graduated from Baylor University with a Bachelor of Science in Informatics. Her technical experience includes Java, C++, JavaScript, TypeScript, Angular, React, React Native, Vue, Flutter and Spring Boot. She has worked for five years at Credera, working on web-based application for C Spire, IFM Restoration, Liberty Oilfield, Orthorix, and Handy.

ONE MORE THING

I was a member of Baylor Latin Dance Society during college and am working as a part time instructor for the Bailart Dance team in the Dallas area.

4