Software Engineering II
Baylor University
August 30, 2018

Table of Contents

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.

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 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 six employees to participate in on-campus activities and regular phone calls 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. Weekly 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, Tempetūrs, wants to connect pet owners with people willing and able to take care of their furry friends during trips, family emergencies, busy work schedules, etc. Sitters will be able to be register their experience and availability, and owners will be matched based on pet type and preferences. You've been hired as a consultant to build out a web application for this project. The client wants your team to create a user-friendly web app for users to register under the appropriate classification, match owners and sitters, and rate sitters who have watched an owner's pets. 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. 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 a thorough walk-through of current complete functionality will be expected.

- **Milestone 1 (due week of Sept 10th):**
    - Have team repository setup
    - Have deployed the template project to a live site on Heroku
    - Have continuous integration setup (a Git commit triggers a build, which triggers a Heroku deployment)
    - 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:**
        - Show the app running live on a Heroku URL
        - Demonstrate that all the pages exist by directly hitting the URLs
        - 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
        - Display the result of the test endpoint service call
- **Milestone 2 (due week of Oct 8th):**
    - Have all the basic pet owner pages implemented – sign-in/registration, owner profile page, home page (at a minimum)
    - Have all the basic pet sitter pages implemented – sign-in/registration, sitter profile page, and sitter scheduling page (at a minimum)
    - In the service layer, be hitting Elasticsearch and be able to create/edit owners, and add/edit pets, create/edit sitters, set sitter availability
    - In the front-end, the pet owner should be able to
        - Register as an owner
        - Log in
        - Add and edit pets
        - Specify preferences for their pets
    - In the front-end, the pet sitter should be able to
        - Register/Log in as a sitter
        - Create a profile
        - Schedule availability
    - Add unit tests in Java layer and add a testing step to the continuous integration deployment pipeline (maintain and update tests moving forward)
    - **Demo:**

        - Demonstrate creating a new pet owner and logging in
        - Demonstrate adding a new pet and providing details about the pet
        - Demonstrate creating a new pet sitter and logging in
        - Demonstrate scheduling availability times
        - Demonstrate deployment, with all tests passing
- **Milestone 3 (due week of Nov 5th):**
    - In the service layer, be able to suggest sitters, schedule and cancel sessions, send notifications, and submit ratings
    - In the front-end, sitters should be able to
        - Get booking notifications and confirm
        - Cancel a booking (for a penalty)
        - Get rating notifications
        - Get reminders about bookings
        - Register as an owner as well
    - In the front-end, owners should be able to
        - Match sitters to owners
        - Search/filter for sitters
        - Schedule bookings
        - Cancel a booking (for a penalty)
        - Rate a sitter
        - Get scheduling reminders
        - Become a sitter as well
    - Update existing and add new unit tests in Java layer
    - **Demo:**
        - Demonstrate sitter suggestions
        - Demonstrate an owner scheduling an appointment
        - Demonstrate a sitter getting notified of an appointment request and confirming it
        - Demonstrate both parties being reminded of the appointment
        - Demonstrate rating a sitter and the sitter being notified
        - Demonstrate a sitter becoming an owner and an owner becoming a sitter
- **Milestone 4 (due at time of Final Presentation):**
    - Enforce security (viewing other people’s pets, non-sitters viewing sitter pages, non-owners viewing owner pages, etc.)
    - Final UI/UX design
    - Bonus features
    - **Demo:**
        - Demonstrate security for various user roles
        - 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
    - Mobile notifications
    - Pet pickup/dropoff
    - Password reset
    - Mobile application
    - Promotions
    - Use of the Alexa service

3.1.3 Deliverables

Every team will be required to complete a weekly deliverable. All deliverables must be completed before the start of your assigned scrum. If the deliverable requires a demo then students must come prepared to show the mentors during their assigned scrum time.

Deliverable	Due Date	Credera Assignment
0	3-Sep-2018	Template project is downloaded, team has a shared repository, Credera mentors have access to this repository
1	17-Sep-2018	Elasticsearch design demo
2	24-Sep-2018	User account creation and login demo
3	1-Oct-2018	End-to-end service demo
4	15-Oct-2018	Sitter suggestions and notifications demo
5	22-Oct-2018	Security features demo
6	12-Nov-2018	UI hardening demo

3.1.4 Schedule

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

Date	Week	Credera Assignment
30-Aug-2018	1	• Class introduction • Project kick-off presentation • Role assignments and requirements clarifications
3-Sep-2018	2	• Labor Day - MONDAY • First scrum meeting • Continued requirements clarification for Milestone 1 • Deliverable 0 Due • Milestone 1 development
10-Sep-2018	3	• Milestone 1 checkpoint during scrum • Continued requirements clarification for Milestone 2 • Begin Milestone 2 development
17-Sep-2018	4	• Milestone 2 development • Deliverable 1 Due
24-Sep-2018	5	• Milestone 2 development • Deliverable 2 Due
1-Oct-2018	6	• Milestone 2 development • Deliverable 3 Due • BUSINESS & STEM Career Fair – THURSDAY
8-Oct-2018	7	• Milestone 2 checkpoint during scrum • Continued requirements clarification for Milestone 3 • Begin Milestone 3 development
15-Oct-2018	8	• Milestone 3 development • Deliverable 4 Due • Fall Break - FRIDAY
22-Oct-2018	9	• Milestone 4 development • Deliverable 5 Due
29-Oct-2018	10	• Homecoming – NO SCRUM • Milestone 3 development
5-Nov-2018	11	• Milestone 3 checkpoint during scrum • Continued requirements clarification for Milestone 4 • Begin Milestone 4 development
12-Nov-2018	12	• Milestone 4 development • Deliverable 6 Due
19-Nov-2018	13	• Thanksgiving Break – NO SCRUM • Milestone 4 development
26-Nov-2018	14	• Final Deliverable (Milestone 4 Demo)

3.2 Office Hours

Office hours will occur weekly. Teams will have a 20-minute slot on Tuesdays or Wednesdays 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:

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

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 of 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:

1. Transaction management – Spring allows for pluggable transaction managers to deal with transactions
2. Inversion of control container – Spring allows for dependency injection, which helps with the configuration and management of Java objects
3. Data access – Easy integration with Hibernate and JDBC
4. Model-View-Controller (MVC) – Spring provides a framework for extending and customizing web applications
5. Messaging – Spring is able to 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.

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 Elasticsearch

Elasticsearch is REST based search tool and data store. It is very fast and reliable, has an API that will allow the service layer to access data easily, and works with many different languages. Elasticsearch is one of the top open-source search tools in the market right now, and has a wide variety of uses.

5. Credera Mentors

John Lutteringer is an Architect at Credera in the Digital Solutions Practice. Having graduated from Baylor University in 2012 with a B.S. in Computer Science, John has over five years of experience in software development ranging from work in ASP.net to Java and C++. On client work, he tackles a variety on interesting challenges related to enhancing and maintaining the online presence of a multi-billion-dollar business. John enjoys canoeing and kayaking, online games, and golf.

Graeme Scruggs is a Consultant in the Integration and Data Services Practice at Credera. Mr. Scruggs serves clients primarily through front-end web and Java development. His skills include: JavaScript, AngularJS, HTML/CSS/SASS, Java, Spring Framework, SQL, Jenkins, and Agile Development. He graduated from Southern Methodist University with a B.S. in Computer Science and a B.S. in Mathematics. Mr. Scruggs previously interned at Sabre, Inc. in the position of Java Developer.

Ali Shan Momin is a Consultant with Credera working in the Integration and Data Services practice. He graduated with a B.S. in Computer Science and Biology from Texas A&M University. Prior to joining Credera, he worked as a Research Assistant at the Texas A&M Health Science Center. Since joining Credera, Ali Shan has worked with projects structured around JavaScript, Java, and RESTful APIs (microservices). He enjoys playing video games, cooking, and programming.

Christopher Blewett is a Consultant in the Integration and Data Services Practice at Credera. He graduated from Baylor University with majors in Bioinformatics and Russian. His technical background is primarily in Java and JavaScript, as well as previous work experience on Android mobile development. At Credera, Christopher has built applications using micro-service architecture to promote global scalability for client solutions. He has also worked on several cloud migration projects, specializing in Amazon Web Services. He enjoys dancing and playing video games.

Christina Flores is a Consultant in the Integration and Data Services Practice at Credera. She graduated from Baylor University with a major in Bioinformatics. Christina’s recent experience includes front-end development for the telecom industry. Before joining Credera, she worked at IBM and holds 7 patent files with the USPTO. In her free time, Christina loves spending time with her cats. She is also a member of several dance teams.

Evan Westermann is a consultant at Credera in the Mobile Solutions Practice. He graduated in May 2017 from Baylor University with a degree in Computer Science. Evan was a four-year member of the Golden Wave Marching Band, and in his free time he plays in a Dallas-based jazz band. This is Evan's first time working with Dr. Song's SWE II class since taking it a few years ago.