



Project Plan Presentation

DSL IDE Test Harness

The Capstone Experience

Team Roosevelt Innovations Knowledge Science

Chase DeVries

Christian Lulaj

Jason Harris

Kristian Rica

Xinghe Zhang

Department of Computer Science and Engineering

Michigan State University

Fall 2022

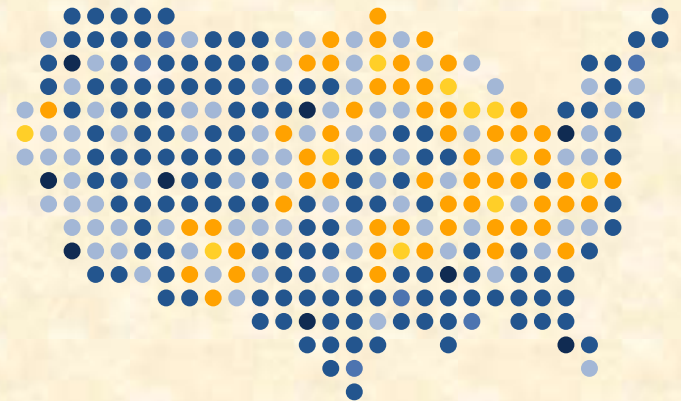


*From Students...
...to Professionals*

Project Sponsor Overview

- Roosevelt Innovations is a software provider for the dental insurance provider Delta Dental
- Located here in Okemos
- Delta Dental initially developed their own software for internal use
- Branched off in January 2022 to develop "company agnostic software" that can be used by any insurance company

Roosevelt
simple. seamless. smart.



Project Functional Specifications

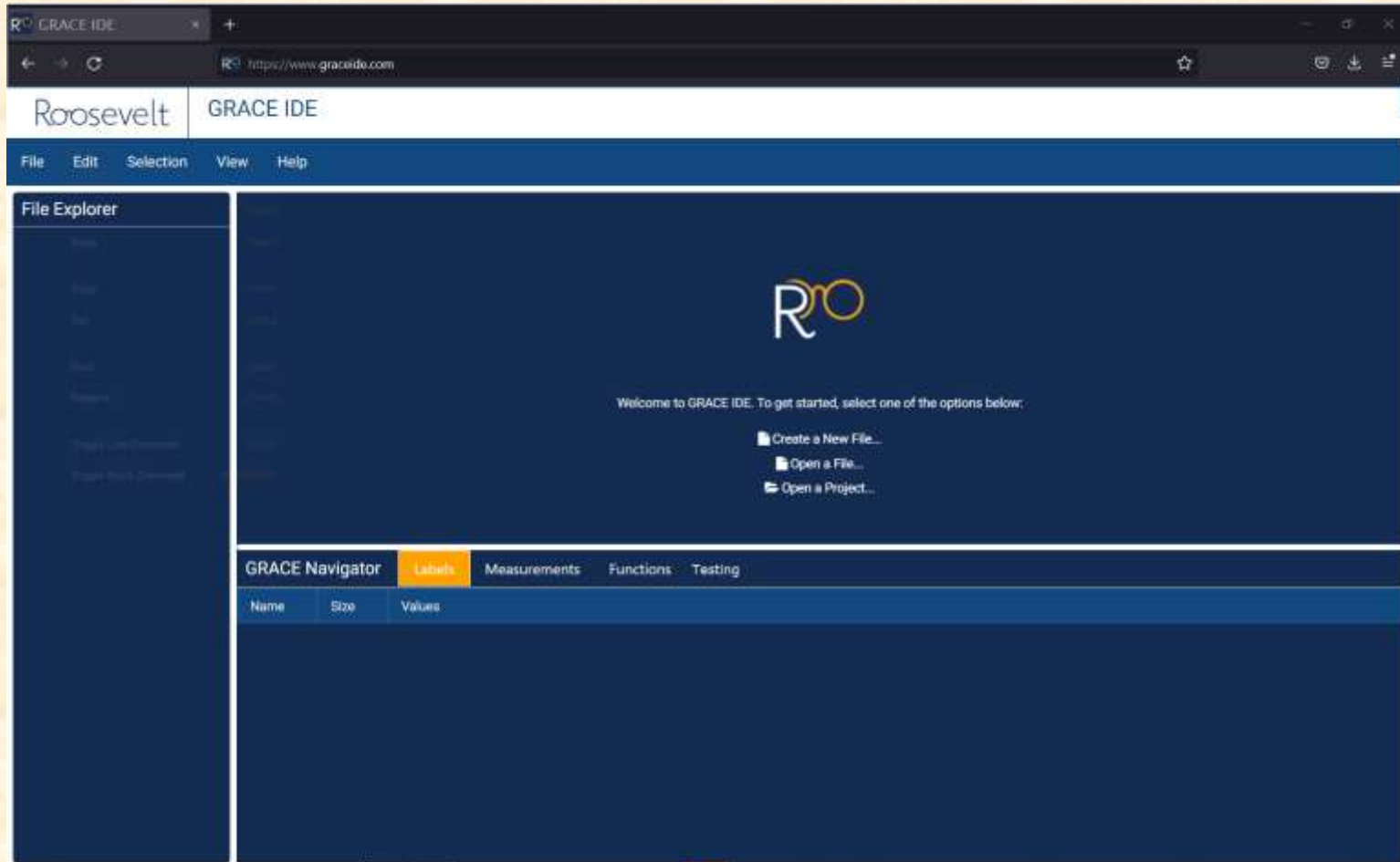
- Allow actuaries to input their own measurements for their calculation models used to perform rate calculations
- Provide the ability for actuaries to perform mock rate calculations using their user defined measurements in their calculation modes
- Improve the proficiency and operation time for actuaries when testing their calculation models
- Decrease the possibility for calculation models to give companies an incorrect rate



Project Design Specifications

- Enhance the existing UI
 - User Interface menu
 - Differentiating User Inputs
 - Testing Screen
 - Results Page

Screen Mockup: Navigation Bar



Screen Mockup: Input Classification

The screenshot displays the ROOSEVELT GRACE IDE interface. The top bar shows the application name and a menu (File, Edit, Selection, View, Help). The left sidebar contains a File Explorer. The main area is a code editor showing a file named `student_grades.grace` with the following code:

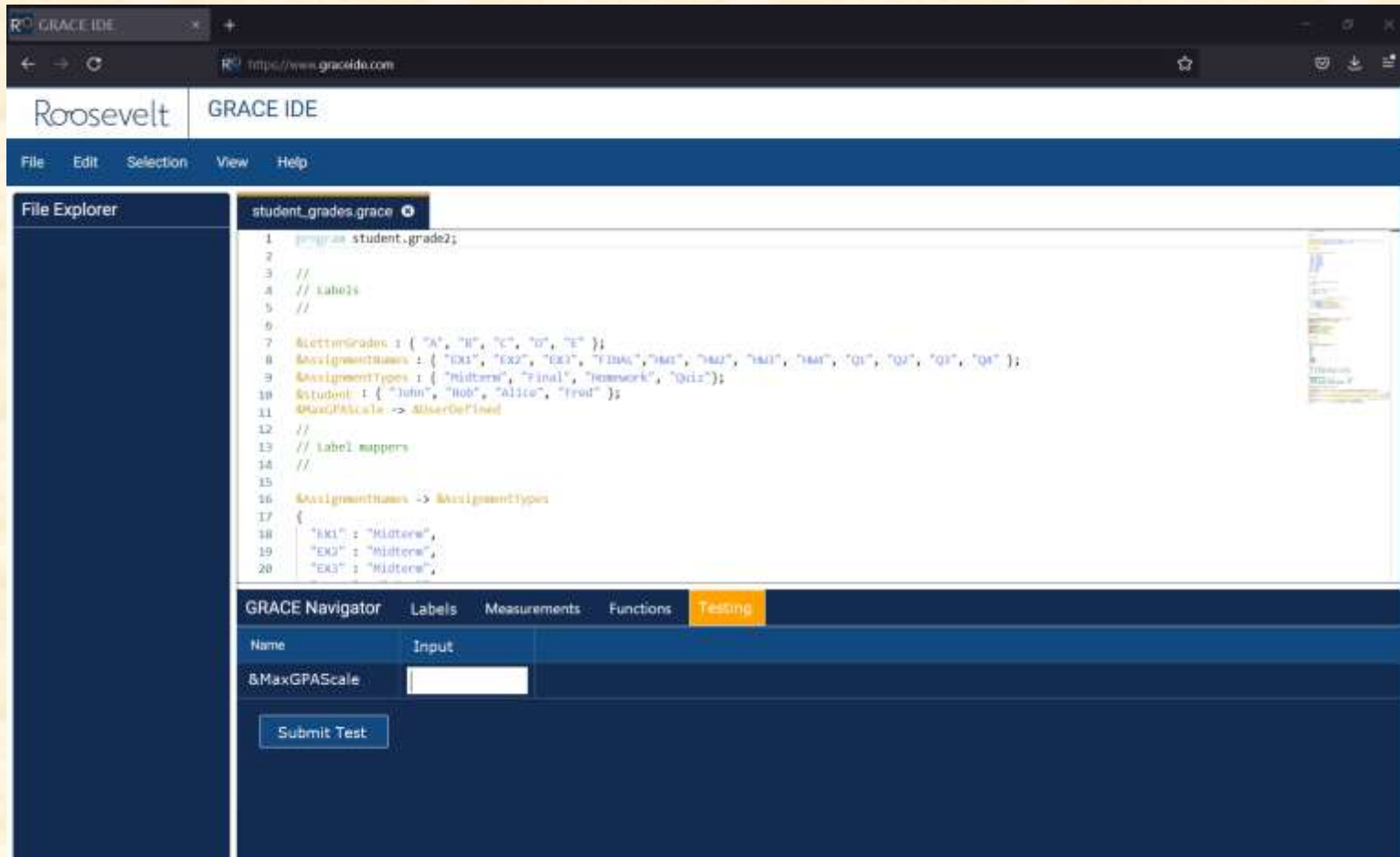
```
1 // student_grades.grace
2
3 //
4 // labels
5 //
6
7 &LetterGrades : { "A", "B", "C", "D", "E" };
8 &AssignmentNames : { "EX1", "EX2", "EX3", "FINAL", "HW1", "HW2", "HW3", "HW4", "Q1", "Q2", "Q3", "Q4" };
9 &AssignmentTypes : { "Midterm", "Final", "Homework", "Quiz" };
10 &Student : { "John", "Bob", "Alice", "Fred" };
11 &MaxGPAScale -> &Undefined
12 //
13 // label mappers
14 //
15
16 &AssignmentNames -> &AssignmentTypes
17 {
18   "EX1" : "Midterm",
19   "EX2" : "Midterm",
20   "EX3" : "Midterm",
```

Below the code editor is the GRACE Navigator table, which is currently showing the 'Labels' tab. The table has four columns: Name, Size, Values, and Input Source.

Name	Size	Values	Input Source
&AssignmentNames	12	EX1, EX2, EX3, FINAL, HW1, HW2, HW3, HW4, Q1, Q2, Q3, Q4	Previously Defined
&AssignmentTypes	4	Midterm, Final, Homework, Quiz	Previously Defined
&LetterGrades	5	A, B, C, D, E	Previously Defined
&Student	4	John, Bob, Alice, Fred	Previously Defined
&MaxGPAScale	1	N/A	User Input



Screen Mockup: User Input



Screen Mockup: Results

The screenshot displays the GRACE IDE web application interface. The top navigation bar includes the Roosevelt logo and the text "GRACE IDE". Below this is a menu with "File", "Edit", "Selection", "View", and "Help". The left sidebar contains a "File Explorer" section. The main area is divided into a code editor and a "GRACE Navigator" section.

The code editor shows the file `student_grades.grace` with the following content:

```
1 program student_grade2;  
2  
3 //  
4 // Labels  
5 //  
6  
7 &letterGrades : { "A", "B", "C", "D", "E" };  
8 &assignmentTimes : { "EX1", "EX2", "EX3", "FINAL", "HW1", "HW2", "HW3", "HW4", "Q1", "Q2", "Q3", "Q4" };  
9 &assignmentTypes : { "Midterm", "Final", "Homework", "Quiz" };  
10 &student : { "John", "Bob", "Alice", "Fred" };  
11 &maxGPAScale -> &fourDefault  
12 //  
13 // label mappers  
14 //  
15  
16 &assignmentTimes -> &assignmentTypes  
17 {  
18   "EX1" : "Midterm",  
19   "EX2" : "Midterm",  
20   "EX3" : "Midterm",
```

The "GRACE Navigator" section has tabs for "Labels", "Measurements", "Functions", and "Testing". The "Testing" tab is active, showing a table with the following data:

StudentName	GPA	MaxGPAScale
John	3.0	4.0
Bob	2.0	4.0
Alice	4.0	4.0
Fred	4.0	4.0

Below the table, there is a "Test Results Above" label and a "Run New Test" button.

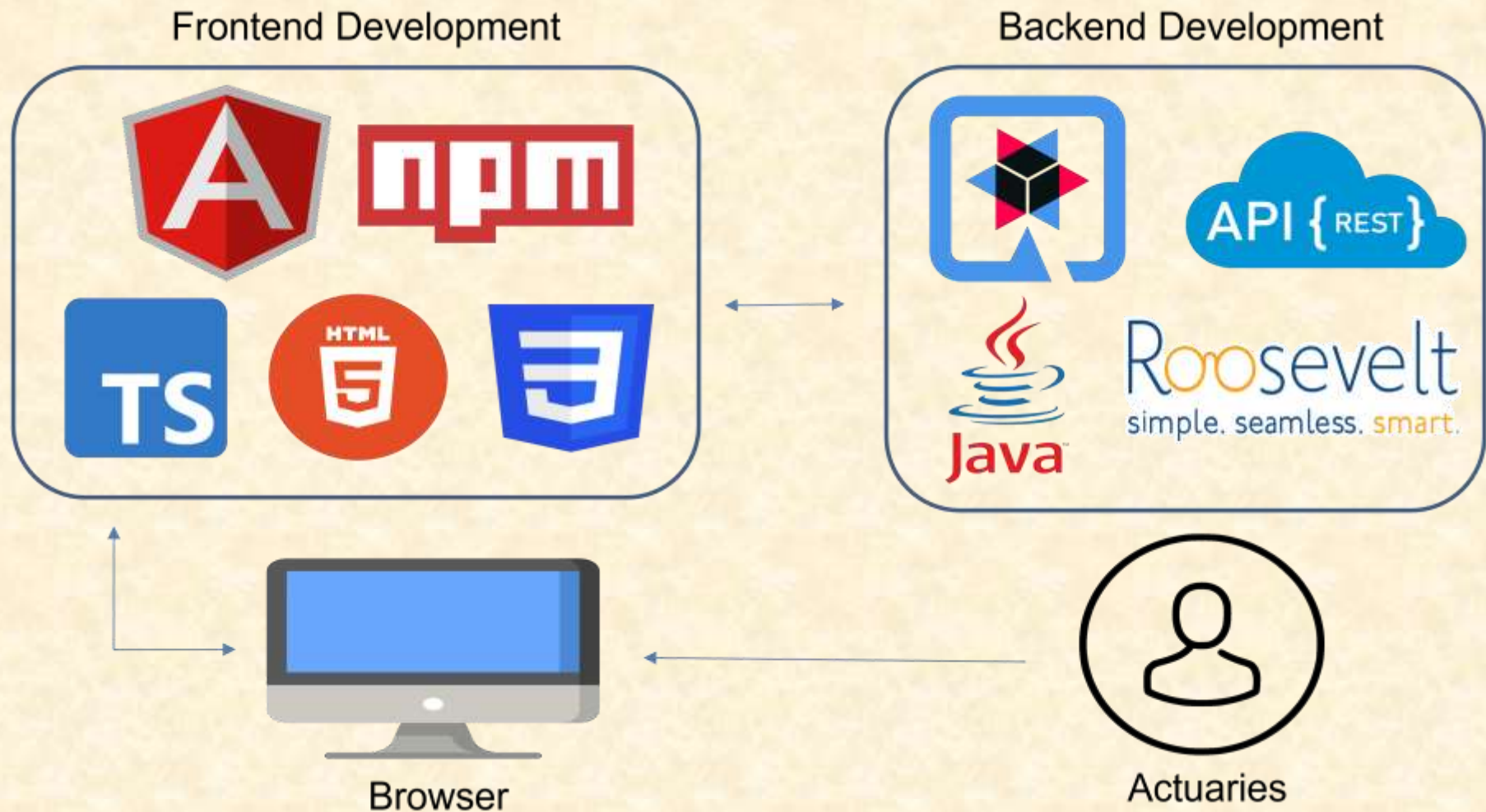


Project Technical Specifications

- Use Angular framework to enhance the existing UI
- Host a Java microservice environment using Quarkus
- Use a REST API to request for a GRACE program from the microservice
- Use ANTLR to parse the GRACE file and populate the UI with user entry fields
- Use rate calculation services to perform mock rate calculations and send that data back to the UI to display to the user



Project System Architecture



Project System Components

- **Frontend Components**

- Angular - is a TypeScript-based free and open-source web application framework. We will be using this framework for dynamically updating frontend components on the user interface.
- HTML/CSS - These components are used for the styling and structure of the user interface.
- NPM - is our package manager and installer.

- **Backend Components**

- GRACE/ANTLR - a powerful parser generator for reading, processing, executing, or translating structured text or binary files.
- Quarkus (Java) - Quarkus optimizes Java and makes it efficient for container, cloud, and serverless environments with memory consumption optimization and a fast response time.
- Restful API - communicate via HTTP requests to perform standard database functions like creating, reading, updating, and deleting records.



Project Risks

- Dynamically generate Angular user interface
 - When the user executes a rate calculation file an Angular page is generated dynamically to request information.
 - Implement an Angular user interface that generates a variable number of labeled input boxes and records the user's input.
- Create a Java microservice to calculate rates for the GRACE IDE
 - When the user inputs sufficient information, the application will send the data to our Java microservice where the rate will be calculated and made available to the user.
 - Create a Java microservice that can perform some arbitrary calculation using values from the IDE and provide the result.
- Create GRACE files for testing purposes using the existing ANTLR grammar
 - The GRACE IDE uses a language that only exists in this application and is not documented as extensively as the languages we have experience with.
 - Create a small functional GRACE file that compiles in the IDE and results in a request to the user for at least one input.
- Design a user interface for users who may not be tech savvy
 - The actuaries using the GRACE IDE may have very limited experience interacting with technical software, and IDE's can present an overwhelming amount of information.
 - Coordinate with actual actuaries at Delta Dental and Roosevelt Innovations to achieve a solution that fits their expectation. Their idea of a good experience is likely different than ours.



Questions?

?

?

?

?

?

?

?

?

?

