Team Dadamapolous Requirements Deliverable

Table of Contents

1. Description
2. List of Teams
3. Formal Requirements
    a. Functional
    b. Non-functional
4. Front-End
5. Text/Video Chat
6. IDE
7. Deadline

Description

Our objective with this project is to create a useable and functional website that allows students to learn how to code in Java or Python via online tutoring sessions and interactions with an actual tutor. Student-tutor interactions will be facilitated by video and text chat and the student will be able to learn how to code with the tutor via a shareable online editor and compiler.

List of Teams

Front End:

- Rachel Stern
- Michael Snajder
- Pat Song
- Huihai Zhou

Text/Video Chat:

- Jarrod Smeyers
- Eli weinberger
- Kah Soon Yap
- David Sevilla

IDE:

- Alex Schlumpf
- Shamroz Shadid
- Stephen Szemis
- Jordan Tantuico
- Sean Trinh

Formal Requirements

Functional:

- Must be able to perform live video and text chat to facilitate effective communication between tutor and student
- Real time code editor and compiler between student and tutor
- At least two switchable programming languages supported (Java and Python)
- Share screen like Discord/Skype
- Have the tutor and student be able to schedule a time for a tutoring session
- Email notifications for meetings
- Dashboard for accessing different parts of the site
- Login/Logout and Registration for Student and Tutor accounts

Non-functional:

- Project will be completed before the end of the semester
- Security and Permissions
    - Encrypted login information
- Need permission to access/disable camera and/or microphone
- Video calling service is fast, reliable, and doesn't require much user intervention
- Students should easily be able to set up a call/meeting with another tutor
- Students and tutors able to sync their Google Calendars to streamline the scheduling of tutoring session (ICS files)
- Should be well designed and easy to navigate (responsive, intuitive UI)
- Scalable and extendable to support several university tutoring and authentication systems (courses in other subjects)

Front-End

User Stories:

Title: User change size of window		
Acceptance Test: sizeWindow	Priority: 3	Story Points: 1
As a student/tutor, I want to be able to minimize/change the size of the video window, so that i have more room for the live editing window on my screen.		

Title: Register account		
Acceptance Test: manageAccount	Priority: 0	Story Points: 1

As a student/tutor, I want to be able to register my student/tutor account, so that i can log into the website and have access to my personal info.

Title: Editing window		
Acceptance Test: editWindow	Priority: 1	Story Points: 3
As a student/tutor, I want the tutor to see my code/see the student's code, so i get/give real time feedback		

Title: Scheduling a Tutor Session		
Acceptance Test: scheduleTutor	Priority: 2	Story Points: 2
As a user, I want to be able to schedule sessions with a tutor so that I can effectively find a time where I can meet or speak with a tutor		

Title: Dashboard Accessibility		
Acceptance Test: accessDashboard	Priority: 3	Story Points: 2
As a user I want to be able to navigate through the dashboard effectively and easily so that i can find what I need on the website with ease		

Title: Security		
Acceptance Test: websiteSecurity	Priority: 0	Story Points: 3
As a user, i wanna be able to securely log in/out, so that no one will be able to access anything they aren't authorized to.		

Title: Remember username		
Acceptance Test: rememberLogin	Priority: 4	Story Points: 1
As a user, I want to have a "remember me" option in the login window, so that the website remembers my username/password and I don't have to type it everytime.		

Basic Flow:

Schedule tutor:

1. Go to the profile of the tutor
2. Click on his calendar
3. Look for openings and send a request to the tutor
4. Tutor can either accept the request or decline
    a. If accepted, the session will appear on both the tutor's calendar and the student's calendar.
    b. If the tutor declines, he must enter a reason for refusal. The student will receive a notification and the reason for decline
5. Either party can cancel a meeting, the other party will receive notification

Login:

1. User enters the site and is able to navigate to the login page.
2. User selects student or tutor.
3. User enters username and password that is securely verified by the database.
4. User gains access to parts of the site that need an account

Register:

1. User enters site and navigates to login page
2. User clicks on "register" button
3. User is prompted to enter his personal information
    a. Username
    b. Password
    c. Email
    d. Option to be Student/Tutor (or both)
4. User clicks "submit"
5. User verifies his/her email
6. User now has new account

UML Diagram:

Individual Responsibilities

- **Website Design:** Rachel Stern
- **Calendar Sharing:** Huihai Zhou
- **Login / Logout (backend):** Pat Song

Video/Text Chat

The video/text chat team will be in charge of setting up the server and handling communications between the tutor and student.

User Stories:

Title: Student and Tutor Enter Room		
Acceptance Test: userEnterRoom	Priority: 2	Story Points: 5
As a user, I want to automatically join a video chat that hosts my upcoming tutoring session appointment so that I don't have to manually call the other involved parties.		

Title: Connect User with Server		
Acceptance Test: connectionEstablished	Priority: 1	Story Points: 2
As an admin, I want the request from the site to go to the server so that the client and server can communicate.		

Basic Flow:

Student and Tutor Enter Room:

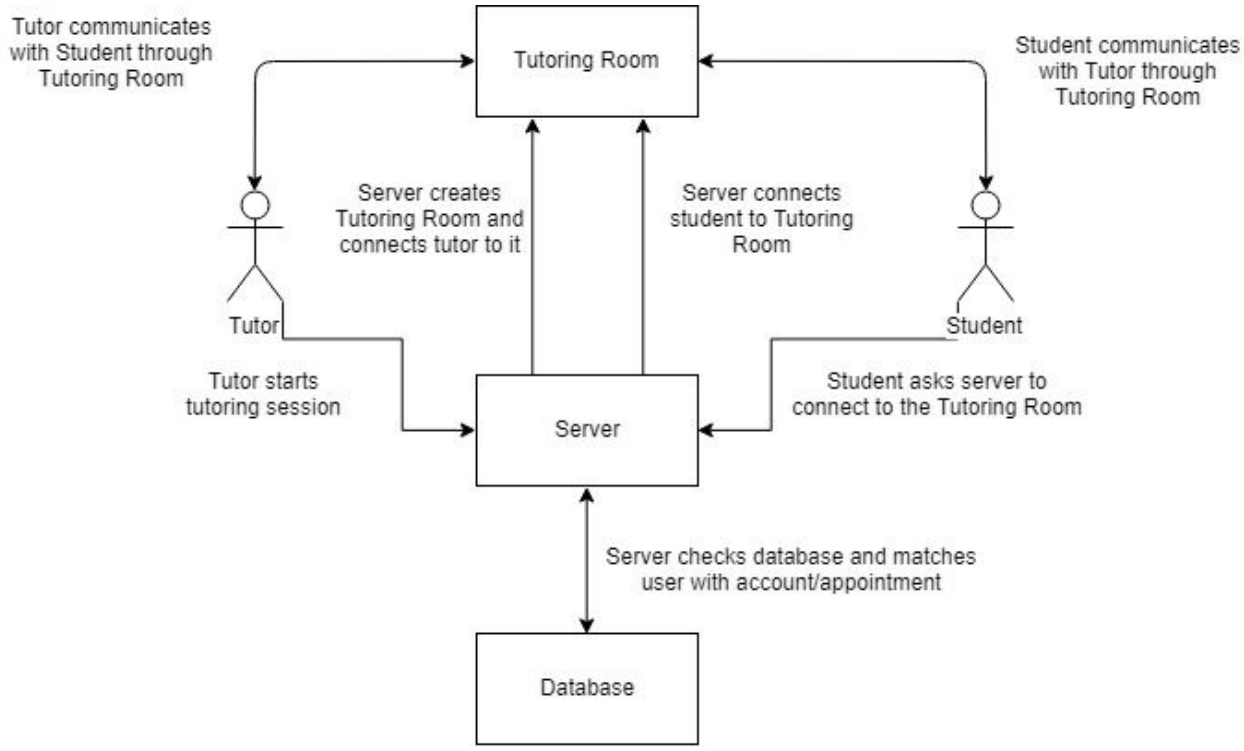
1. User logs into the site
2. Entering Session
    a. Tutor: Receives session notification and clicks "START" to start the room.
    b. Tutee: Receives session notification and clicks "JOIN" button to join the existing room.
3. If the user has disabled audio and voice by default, they receive a pop up asking for the site to use their microphone and webcam
4. Once student has entered the session, tutor and student can code together on the shared compiler, can see and hear each other on the video call, and can end the session

Connect User With Server:

1. User sends request to server

    a. If the user is a tutor, user presses “START SESSION” and sends a request to the server to create a tutoring session
    b. If the user is a student, user waits for a notification that his/her tutoring session is live and presses “JOIN SESSION” and sends a request to the server to join the session
2. Server creates server
3. Server responds to user
    a. If user is a tutor, server connects user to the session, notifies the user that he/she is connected, and sends a notification to the respective student that session is live (checks for the right student by sending a query to the database)
    b. If user is a student, server connects user to the session and notifies student he/she is connected

UML Diagram:



Individual Responsibilities

- **Video chat front end** - David Sevilla, Eli weinberger, Jarrod Smeyers
- **Video chat server integration** - All
- **Server** - Kah Soon Yap

IDE

The IDE team will be responsible for creating a real time code editor and compiler between students and tutors. They will be able to switch between at least two programming languages, such as Java and Python.

User Stories:

Title: Student Writes Code		
Acceptance Test: studWrite	Priority: 1	Story Points: 2.5
As a student, I want to be able to write code that the tutor can see and code that can compile in one of two programming languages (Java or Python).		

Title: Teacher Writes Code		
Acceptance Test: teachWrite	Priority: 1	Story Points: 2.5
As a tutor, I want to be able to demonstrate how to write code to the student in one of two programming languages (Java or Python). I would also like for the code to be able to compile.		

Title: Student Download Code		
Acceptance Test: downloadFile	Priority: 2	Story Points: 2
As a user, I want to be able to download the code that is written to the appropriate file type. (ie if I coded in Python, I want to be able to download the code to a .py file. If I coded in Java, I want to be able to download the code to a .java file).		

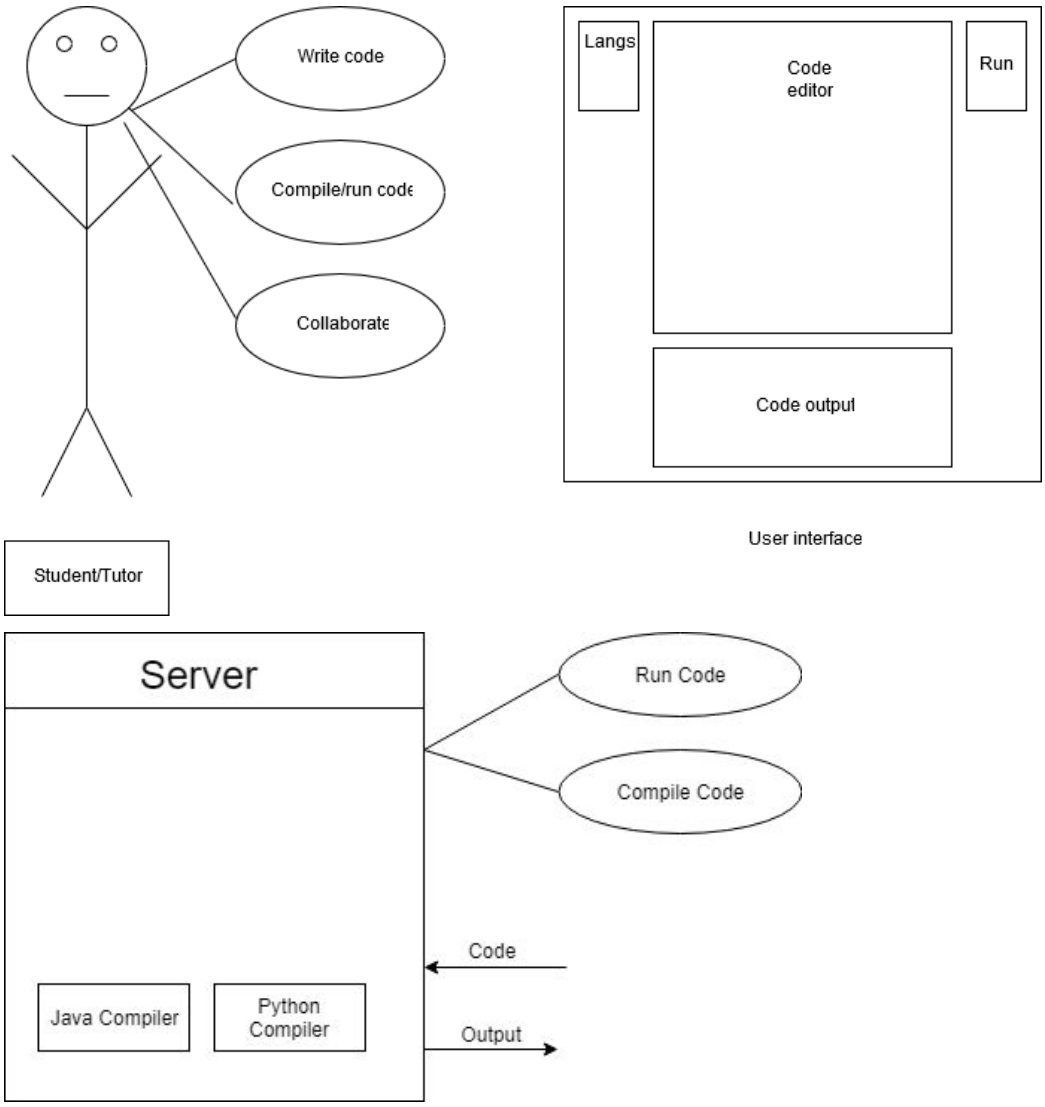
Title: Teacher Upload Code		
Acceptance Test: uploadFile	Priority: 4	Story Points: 2
As a user, I want to be able to upload some of my own code for the student to be able to see and make edits too.		

Basic Flow:

1. The tutor selects the programming language of choice.
2. Tutor/student starts coding.
3. Either the tutor or student clicks the compile button.

4. The code either compiles or raises an error that is output to a console.
5. The process continues until the tutoring session ends.
6. Student can download code they wrote so they can look at it later

UML Diagram:



Individual Responsibilities:

- **Student writing code** - Shamroz Shadid, Jordan Tantuico
- **Teacher writing code** - Stephen Szemis
- **Student downloading code** - Alex Schlumpf
- **Teacher uploading code** - Sean Trinh

Deadlines:

March 1st:

a. Completed research on software tools

    b. Completed research on databases
    c. Complete research on video platform
    d. Login tools
1. March 8th:
    a. Working editor and compiler on a local machine
    b. Rough plan (layout) for website completed
    c. Video chat working using separate login
2. March 15th:
    a. Be able to download and upload code locally
3. March 22th:
    a. Set up appropriate servers
    b. Users should be able to log in and out
4. March 29th:
    a. Establish connection between students, teachers, and servers
    b. Roles assigned to accounts (accounted customized for roles)
5. April 5th:
    a. Be able to download and upload code from either connection from the server
    b. Format calendar for time sharing
    c. Video chat integrated with tutor account
6. April 12th:
    a. Optimization
7. April 19th:
    a. Integration with other features of the project
8. April 26th:
    a. Rigorous testing completed
9. May 3rd:
    a. Project should be completed and ready to send out to users

We pledge my honor that we have abided by the Stevens Honor System