

Team: Lauren Mitchell
Peilun Zhang
Alan Moy
Ian Char
Taylor Andrews

Title: CodingCareers

Project Summary: CodingCareers is a collection of browser-based interactive coding lessons for children ages 11-14. Our goal is to expose children to programming in python, as well as the opportunities coding has to offer. We will have different levels that can be reached, with different careers on each level. We plan on rotating between an interactive lesson plan and real world problem solving. One level will involve a hospital or otherwise healthcare related theme to show that computer scientists have a part to play in that field. CodingCareers is designed to be easy to use. We will not require users to login to experience CodingCareers.

Project Requirements:

Business Requirements		
ID	Requirement	Priority (5 being the highest, 1 being the lowest)
B-01	Users gain virtual numeric Career Points associated with their account as rewards.	3

User Requirements		
ID	Requirement	Priority (5 is highest)
U-01	As a user, I need to be able to login with my username and password so that I can keep track of my progress.	4
U-02	As a user, I need to be able to submit my code snippet for a particular exercise so that I can receive a feedback.	5
U-03	As a user, I need to be able to see all of the available lessons so I can navigate to the one that I am interested in.	3
U-04	As a user, I need to be able to log out of the system in case I want to log in as somebody else.	4
U-05	As a user, I need to have some way to input my Python code into a form so that I can submit it.	5
U-06	As a user, I need to be able to find out what the site is and how it works so I can decide if I want to register.	2
U-07	As a user, I need to have a way of creating a new account so that I can use the site.	4
U-08	As a user, I should be presented with the option of moving on to the next task upon completion so that I can easily progress through the levels.	4
U-09	As a user, I need to be able to select my character so that I can be represented in CodingCareers.	1
U-10	As a user, I need to be able to reset my code with a button to go to the initial state of each stage.	1
U-11	As a user, I need to be given a hint to help me if I'm stuck for more than 5 minutes on a level.	1
U-12	As a user, I need to be able to view my overall progress in order to see how much I have learned.	3

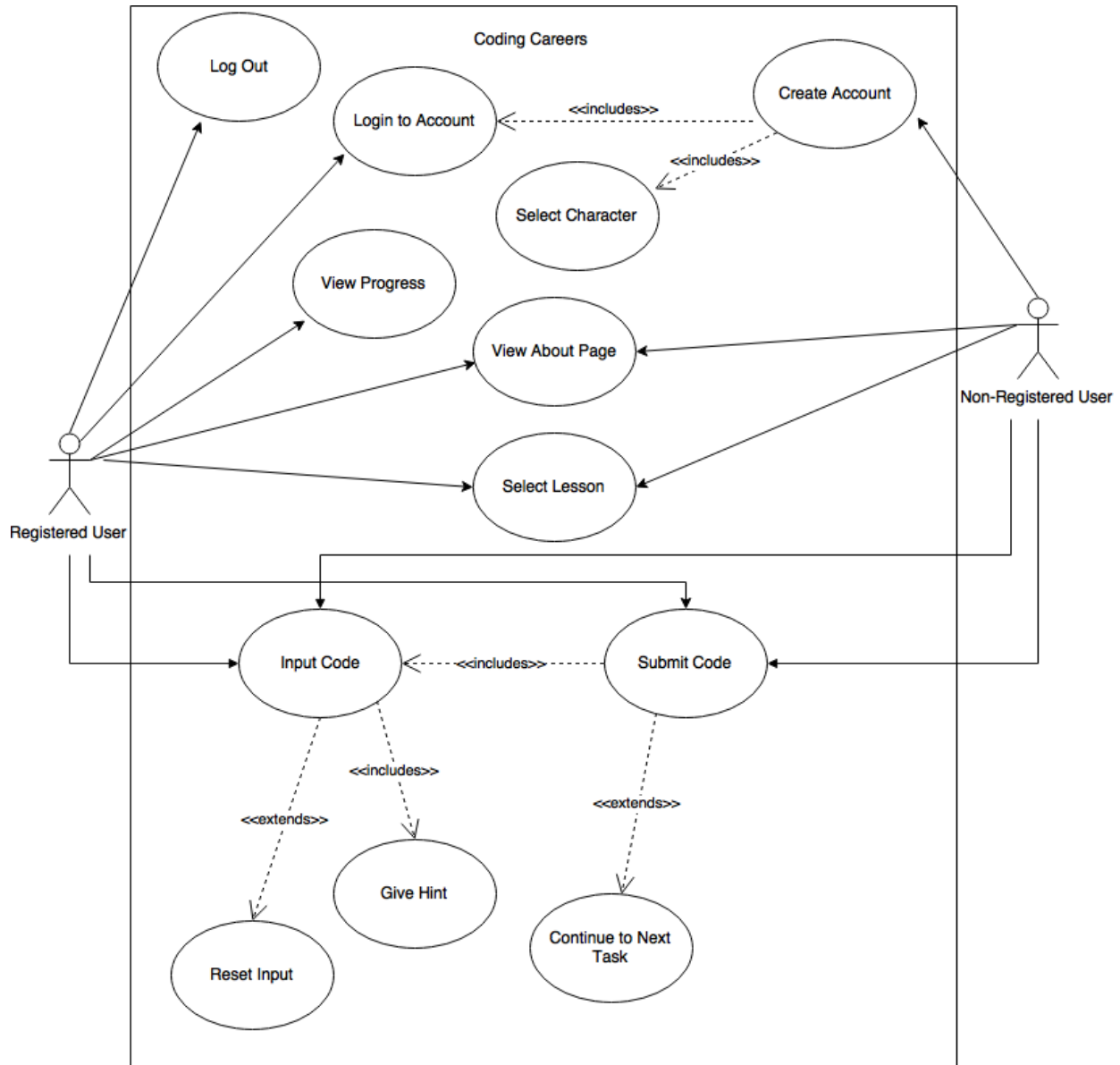
Functional Requirements		
ID	Requirement	Priority (5 is highest)
F-01	The system should assign reward points for every test case passed.	3
F-02	Upon login, a token must be generated for the client to identify the user's session.	4
F-03	The system should assign bonus reward points when the user logs in 5 days in a row.	3

Non-Functional Requirements		
ID	Requirement	Priority (5 is highest)
NF-01	Security: The database needs to store passwords securely so account data cannot be stolen.	5
NF-02	Performance: The system should spend less than 10 seconds loading each level.	2
NF-04	Security: The system needs to validate the user's code in a safe environment so no injection occurs into the system.	5
NF-05	Performance: The system should spend no more than 10 seconds validating the user's code.	3

Use Cases:

Actors: Registered User, Non-registered User

Use Case Overview: Since both Registered and Non-registered users can input and submit code they both have arrows to these use cases.



Use Case Documents:

Use Case ID:	UC-01
Use Case Name:	Login to Account
Description:	Users need to be able to login with their username and password so they can keep track of their progress and achievements in the CodingCareers learning experience.

Actors:	Registered User		
Pre-Conditions:	User is on the CodingCareers sign-in webpage, and the user is not logged into their account.		
Post-Conditions:	User is logged into their account, user is viewing the lesson selection page.		
Frequency of Use:	Every time the user wants to access their account.		
Flow of Events:		Actor Action	System Response
	1	Enter username and password.	
	2	Hit login button.	Validate the user's credentials.
	3		Redirect the browser to the lessons page.
Variations:	1. If the user enters incorrect login data and is prompted to retry.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-02
Use Case Name:	Submit Code
Description:	The user submits their written code so its correctness can be evaluated by the system. This allows the user to see the progress they have made in the current level.

Actors:	Registered and unregistered users.		
Pre-Conditions:	The user selected a level, and added some amount of code that can be evaluated.		
Post-Conditions:	Information about the results of the user's code will be displayed. This includes score and failed trials.		
Frequency of Use:	Every time the user wishes to check the accuracy of their solutions to CodingCareers challenges.		
Flow of Events:		Actor Action	System Response
	1	Press the submit button.	Execute the python code.
	2		Assess which tests the player's code passed.
	3		Display the results of the tests to the user.
Variations:	1. The user submits code that cannot be interpreted by python. An error message is displayed.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-03
Use Case Name:	Select Lessons
Description:	Users need to be able to view the lessons, and a short description about each lesson, in order to select the lesson they want to take.

Actors:	Registered and non-registered user		
Pre-Conditions:	The user is on the CodingCareers website.		
Post-Conditions:	The user is viewing the lesson they selected. The user's previous information is loaded.		
Frequency of Use:	Many times during a CodingCareers session.		
Flow of Events:		Actor Action	System Response
	1	Select the lessons tab.	Redirect the browser to the lessons page which indicates the lessons' order.
	2	Hover over a lesson.	Load and display lesson information. Display pervious high score.
	3	Select the desired lesson by clicking "Start Lesson".	Load the desired lesson and redirects to the appropriate page.
Variations:	2. The user hovers over a lesson while they are not logged in so the system does not display previous attempt information. 3. The user is logged in and selects a lesson they have already attempted. The system displays their past score and most recent code attempt.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-04
Use Case Name:	Log Out
Description:	The user must be able to log out of their account.

Actors:	Registered user		
Pre-Conditions:	The user is on logged into CodingCareers.		
Post-Conditions:	The user is logged out of CodingCareers.		
Frequency of Use:	Every time the user is logged in and desires to stop using CodingCareers.		
Flow of Events:		Actor Action	System Response
	1	Press the logout button.	Log the user out.
	2		Redirect the user to the login page.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-05
Use Case Name:	Input Code
Description:	The user enters the code that solves the specific task into the editor on the webpage.

Actors:	Registered and non-registered user		
Pre-Conditions:	The user has selected a specific level to code for.		
Post-Conditions:	The text the user inputs is highlighted in the input field.		
Frequency of Use:	Many times during a CodingCareers session.		
Flow of Events:		Actor Action	System Response
	1	Type code into the text editor.	Highlight the code based on Python syntax.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-06
Use Case Name:	View About Page
Description:	Potential CodingCareers users can navigate to an about page on the website that has information about the purpose of CodingCareers.

Actors:	Registered and non-registered users		
Pre-Conditions:	User is on the CodingCareers website.		
Post-Conditions:	User is looking at the about page and learning about the purpose of the system.		
Frequency of Use:	Infrequent, likely only several visits during first uses of the system.		
Flow of Events:		Actor Action	System Response
	1	Click on the 'About' tab.	Browser navigates to About page.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-07
Use Case Name:	Create Account
Description:	Any user that wants to should be able to make an account to track and save their progress in CodingCareers.

Actors:	Non-registered user		
Pre-Conditions:	User is not logged in.		
Post-Conditions:	The user's account data now exists in the database and the user is logged into their new account.		
Frequency of Use:	Once per account created.		
Flow of Events:		Actor Action	System Response
	1	Enter desired username and password.	Store username and password in database.
	2		Redirect browser to the create character page.
Variations:	1. The user enters a username that is already taken, so the system does not create them an account.		
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-08
Use Case Name:	Continue to Next Task
Description:	After completing a level, a user can click a button to take them to the next level.

Actors:	Registered and non-registered users		
Pre-Conditions:	User has progress in CodingCareers. User has completed a level and the system has presented them with a screen summarizing their progress.		
Post-Conditions:	User is looking at the level following the one they just completed and can begin learning immediately.		
Frequency of Use:	Many times during a CodingCareers session.		
Flow of Events:		Flow of Events:	
	1	Click on the button labeled 'Next'.	Load the next level.
	2		Redirect the browser to the next level.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-09
Use Case Name:	Select Character
Description:	Users can select their character so that they can be represented in CodingCareers. There is some degree of character customization.

Actors:	Registered users		
Pre-Conditions:	The user just created account credentials which save been saved in the CodingCareers database. The system has redirected the user to the select character page.		
Post-Conditions:	The new user has a character bound with their account.		
Frequency of Use:	Once for each new account.		
Flow of Events:		Actor Action	System Response
	1	Select character head.	Save the user's desired head choice to the database.
	2	Select character body.	Save the user's desired body choice to the database.
	3	Select character legs.	Save the user's desired legs choice to the database.
	4		Redirect the browser to the lessons page.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-10
Use Case Name:	Reset Input
Description:	While working on a level, the user can click a button to reset the level as if it was just loaded.

Actors:	Registered and non-registered users		
Pre-Conditions:	User is in progress in a level. User has entered some amount of their own code.		
Post-Conditions:	All code entered is deleted. User is looking at a clean slate.		
Frequency of Use:	Many times during a CodingCareers session.		
Flow of Events:		Actor Action	System Response
	1	Click on the button labeled 'Reset'.	Reload the current level, deleting all code the user has entered to this point.
Variations:			
Notes and Issues:			
Developer Notes:			

Use Case ID:	UC-11
Use Case Name:	Give Hint
Description:	User needs to be given a hint to assist their current task after they have spent more than five minutes on a level.

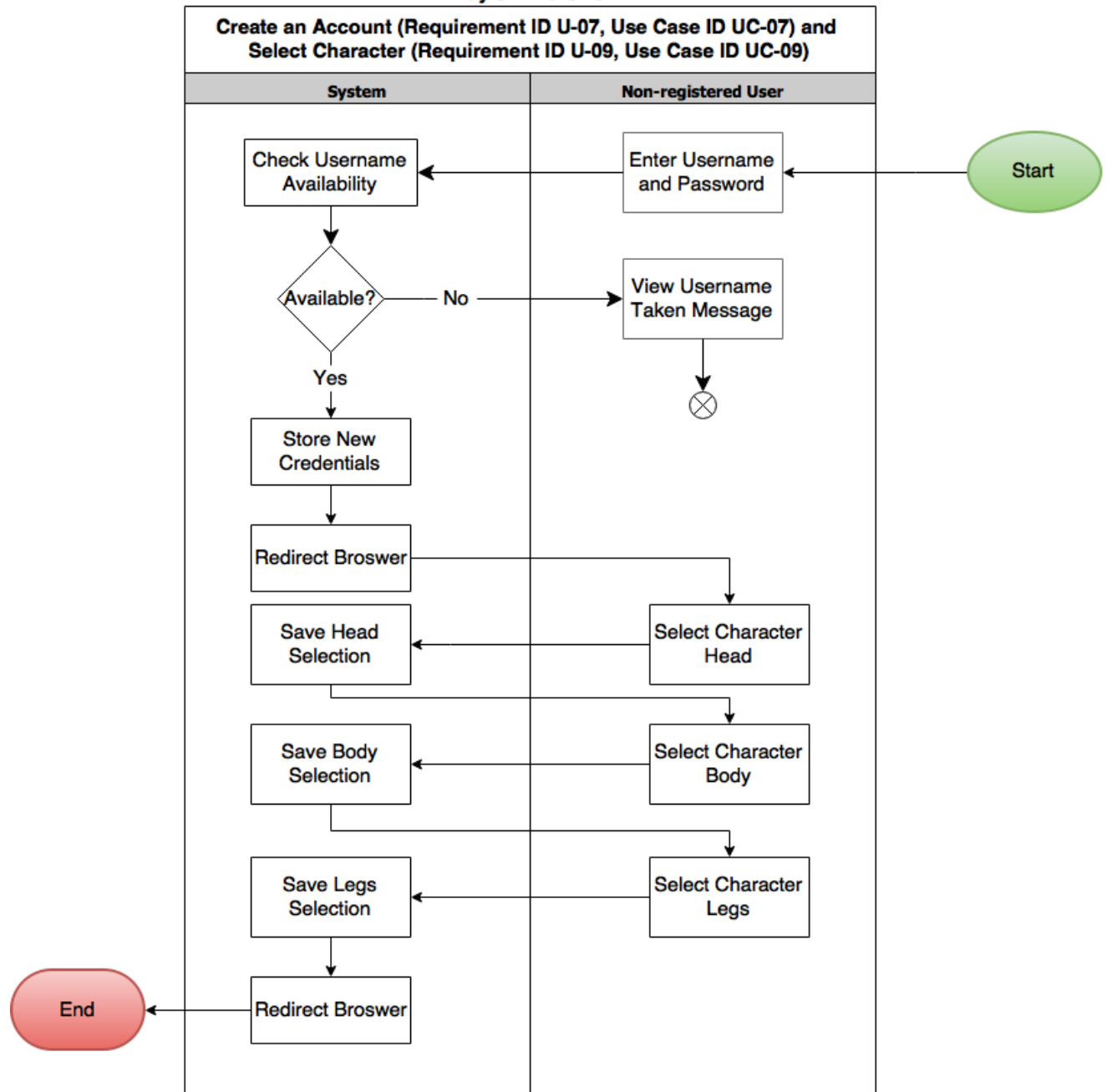
Actors:	Registered and non-registered users		
Pre-Conditions:	The user has selected a level and is currently working on a level on the input.		
Post-Conditions:	Information assisting the user will the current level will be displayed on a pop-up screen after 5 minutes of being on that level.		
Frequency of Use:	After 5 minutes on same level.		
Flow of Events:		Actor Action	System Response
	1	Select level.	Timer begins.
	2		After 5 minutes, display a hint on a pop up screen.
	3		If no input is received, display another hint after 5 more minutes (10 minutes in total run time).
Variations:			
Notes and Issues:			
Developer Notes:	Allow for two hints only, one every five minutes.		

Use Case ID:	UC-12
Use Case Name:	View Progress
Description:	The user needs to be able to view their progress through CodingCareers, what lessons they have completed, and their scores on these lessons.

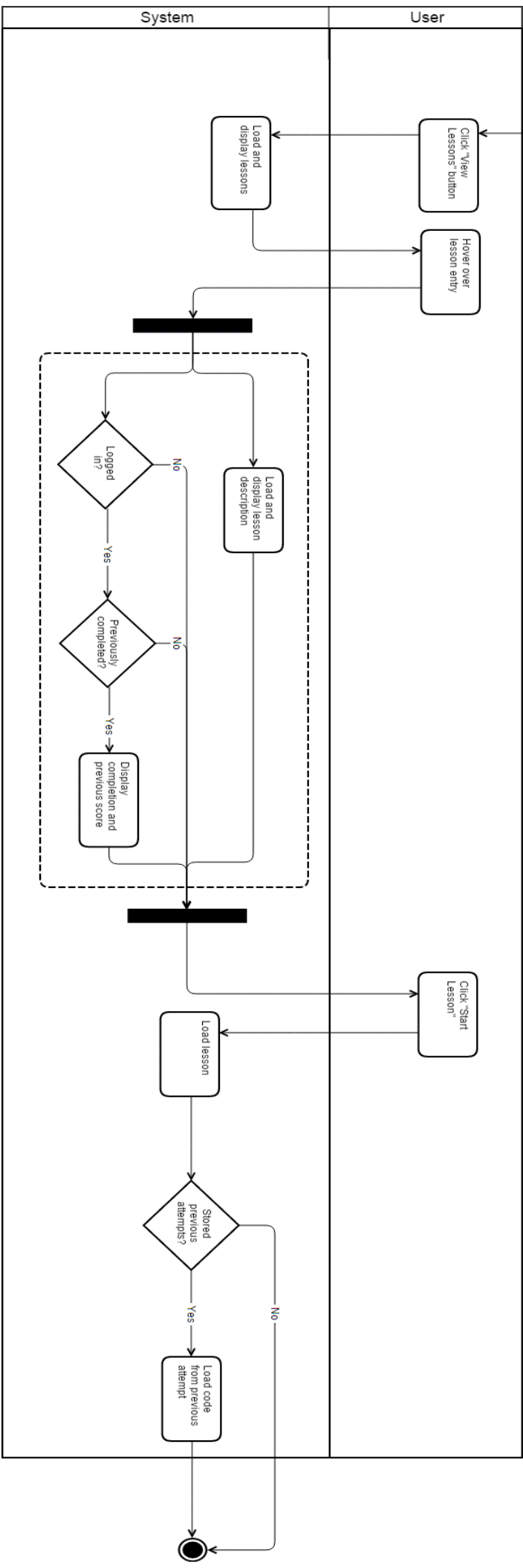
Actors:	Registered user		
Pre-Conditions:	User is logged in.		
Post-Conditions:	User sees their progress.		
Frequency of Use:	Many times during a CodingCareers session.		
Flow of Events:		Actor Action	System Response
	1	Click on the 'Profile' tab.	Query the database to determine user's ids and other information.
	2		Display the profile page of the user.
	3	Click on the 'Progress' button.	Query the database to determine past user's scores and completed lessons.
	4		Display the progress page of the user.
Variations:			
Notes and Issues:			
Developer Notes:			

Activity Diagrams:

Taylor Andrews



Alan Moy
Requirement 03
Use Case U-03
View available lessons and navigate to one

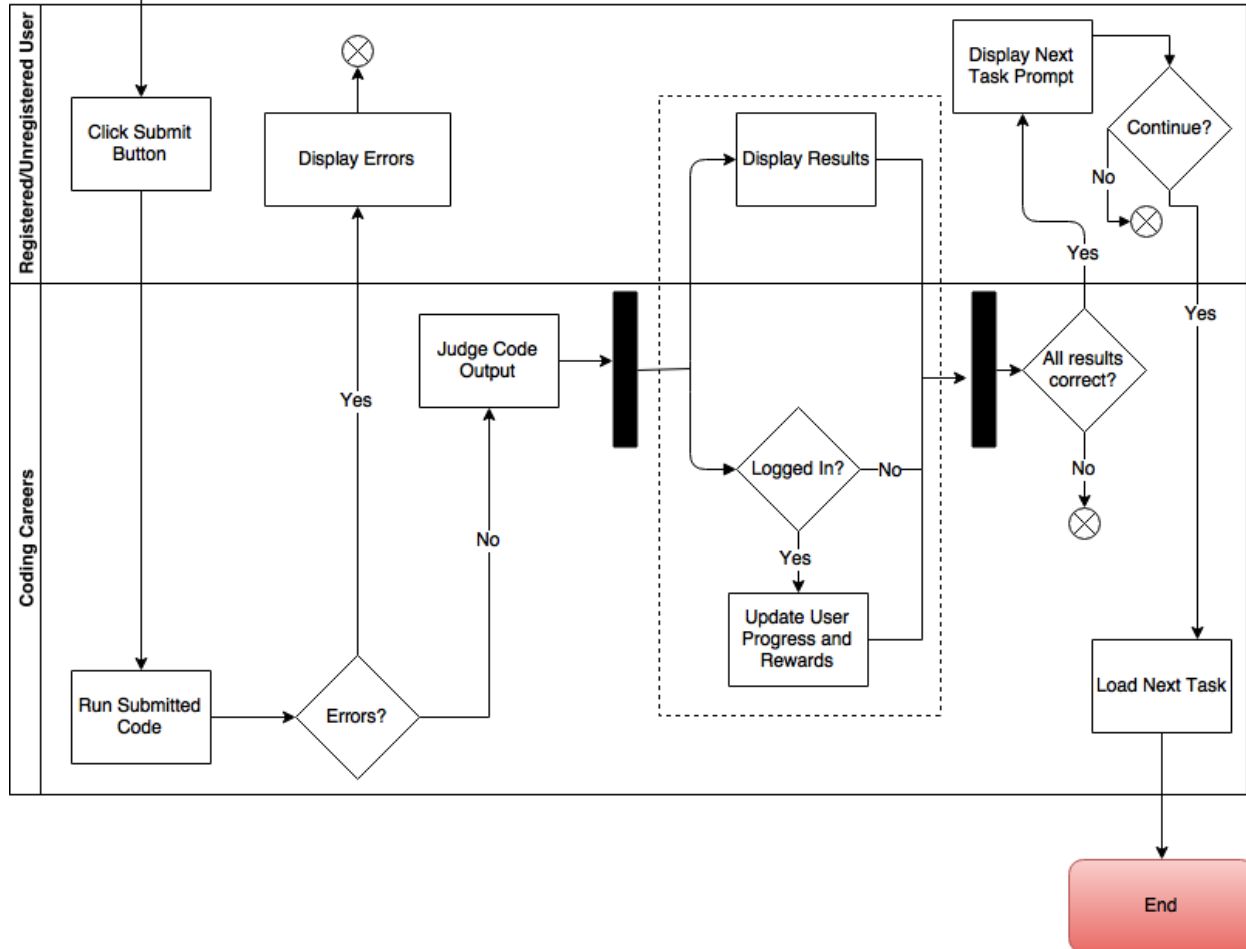


Ian Char

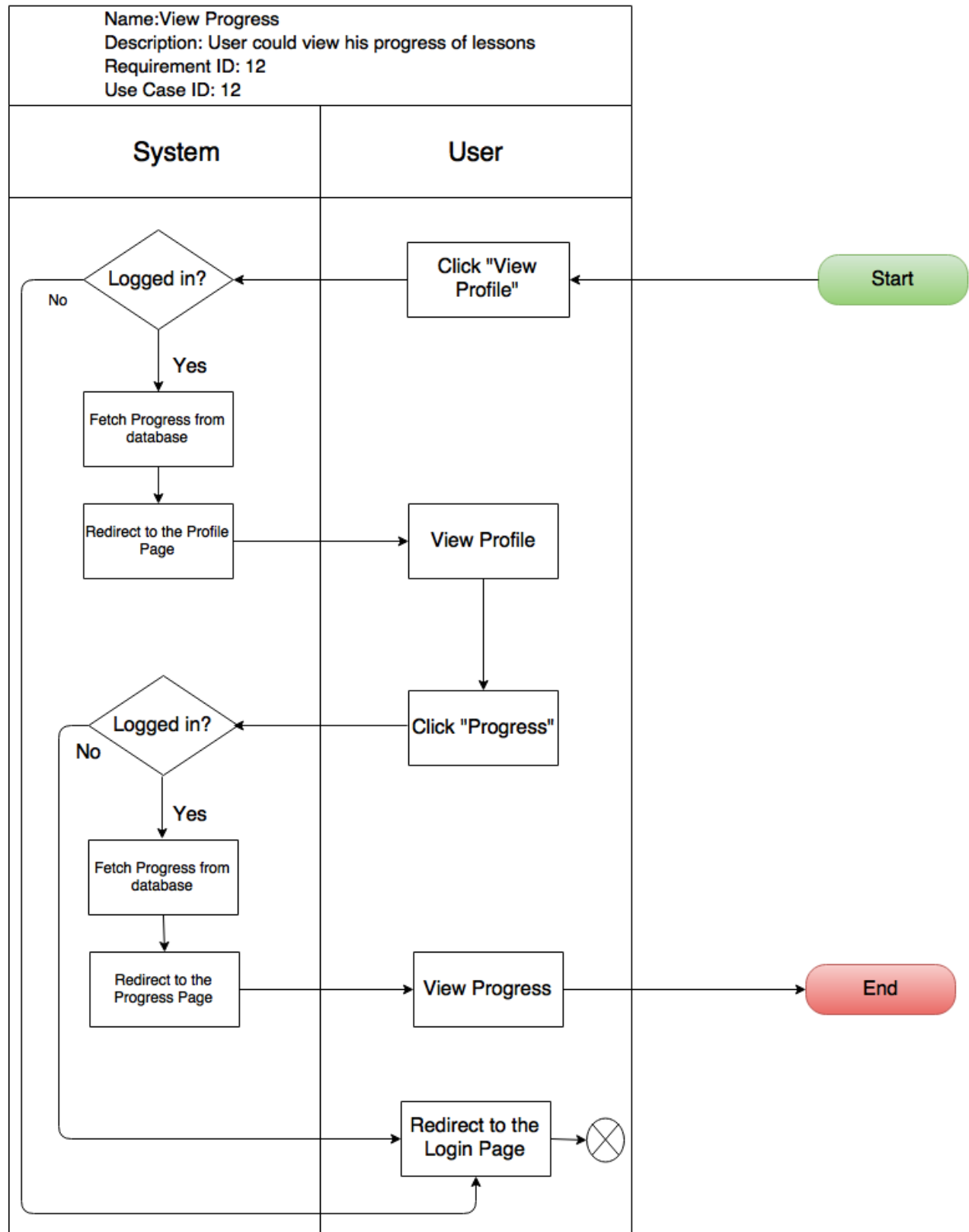
Requirement IDs: U-02, U-08, F-01

Use Case IDs: UC-02, UC-08

Description: This activity diagram outlines what happens when the user (registered or unregistered) submits their code for a particular task.



Peilun Zhang

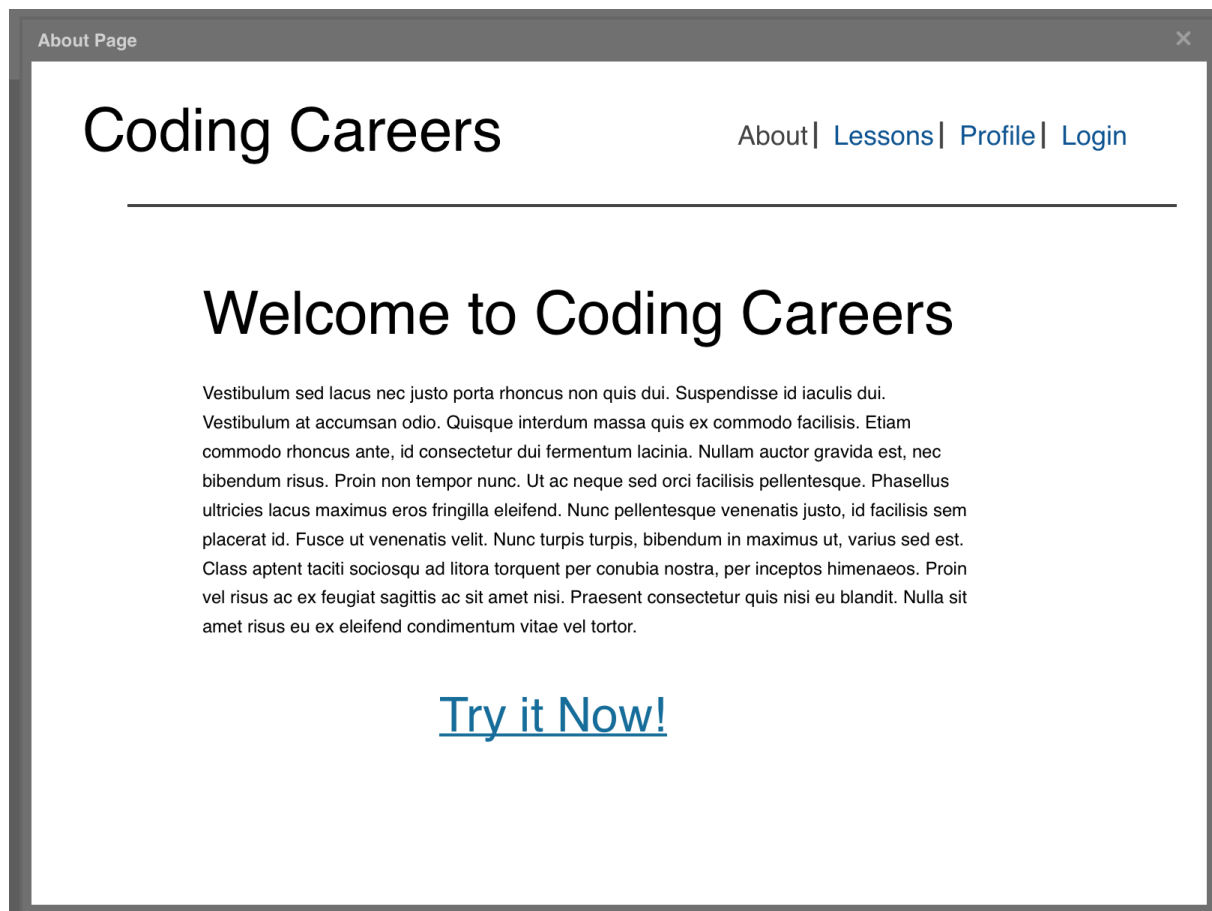


Data Storage: MySQL

Classes:

- Database class to abstract the interactions with MySQL so we could change methodology easily.
- UserManager class to store information about the user, including user ids, rewards, and progress.
- TaskManager class to store task information including instructions of tasks and test cases.

UI Mockups: When the user loads the coding careers webpage, they will be directed to a home page that has an option to sign up. At the top of every page, there is a header in which the user can directly read more about the site by clicking “About”, a “Lessons” page to show all the lessons and tutorials to select from, a “Profile” page where the logged in user can track their progress in the lessons, and a “Login” page where the user will be directed to a page to login to their account. Once the user creates an account, they will be prompted to create a character avatar.



Coding Careers

[About](#) | [Lessons](#) | [Profile](#) | [Login](#)

Select a Character



Coding Careers

[About](#) | [Lessons](#) | [Profile](#) | [Login](#)

Welcome to Coding Careers

The interactive learning experience for students of computer science.

[Learn More!](#)

Coding Careers

[About](#) | [Lessons](#) | [Profile](#) | [Login](#)

Instructions go here

[Save](#)[Run](#)

Input Code

[Clear](#)

Grade:

Coding Careers

[About](#) | [Lessons](#) | [Profile](#) | [Login](#)

Lesson 1: Helping through Code

Sublesson 1

Sublesson 2

Sublesson 3

Sublesson 4

Lesson 2: NASA

Lesson 3: The FBI

Lesson 4: IOS- Developer

Coding Careers

[About](#) | [Lessons](#) | [Profile](#) | [Login](#)

If you already have an account, login here!

Coding Careers

[About](#) | [Lessons](#) | [Profile](#) | [Logout](#)

Your Rewards

Congradulations on task 1



Your Progress

Lesson

Score

Completed

Lesson 1

100%



Lesson 2

100%



Coding Careers

About | [Lessons](#) | [Profile](#) | [Logout](#)

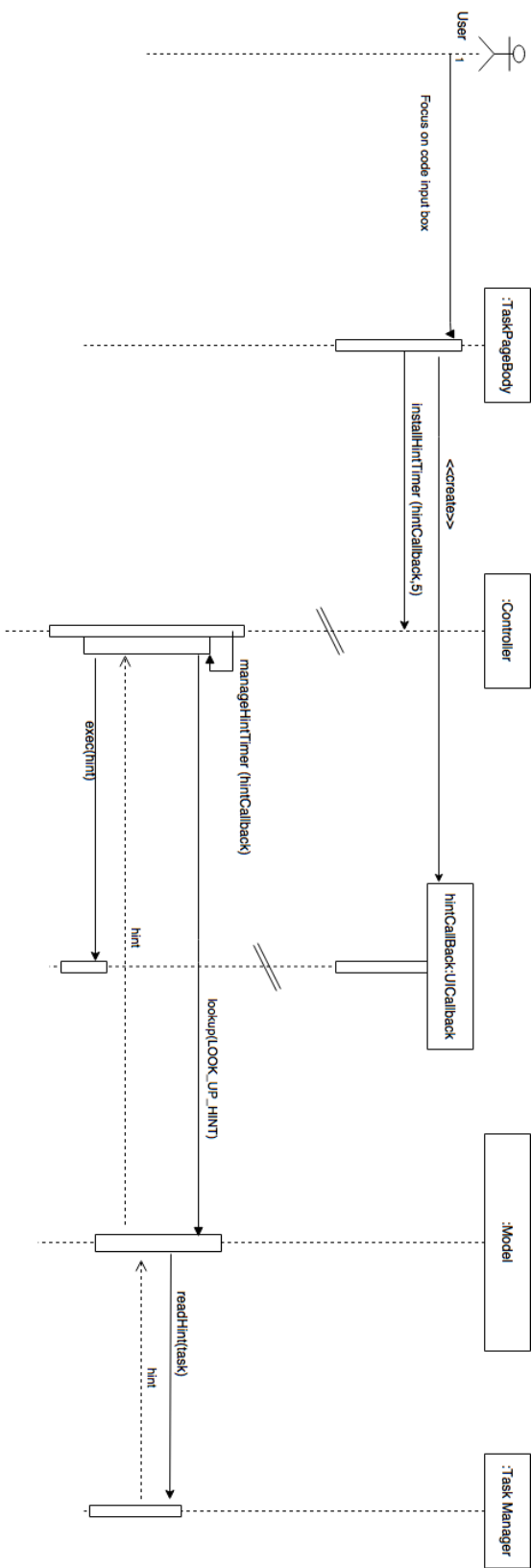
Welcome back, Fred!



Maecenas pretium justo at nunc imperdiet, id bibendum ipsum feugiat. Nunc vel nisi at risus rutrum venenatis eu a est. Nam suscipit pretium placerat. Quisque aliquet dictum sagittis. Mauris felis neque, imperdiet consectetur iaculis eu.

[View Progress](#)

User Interactions:

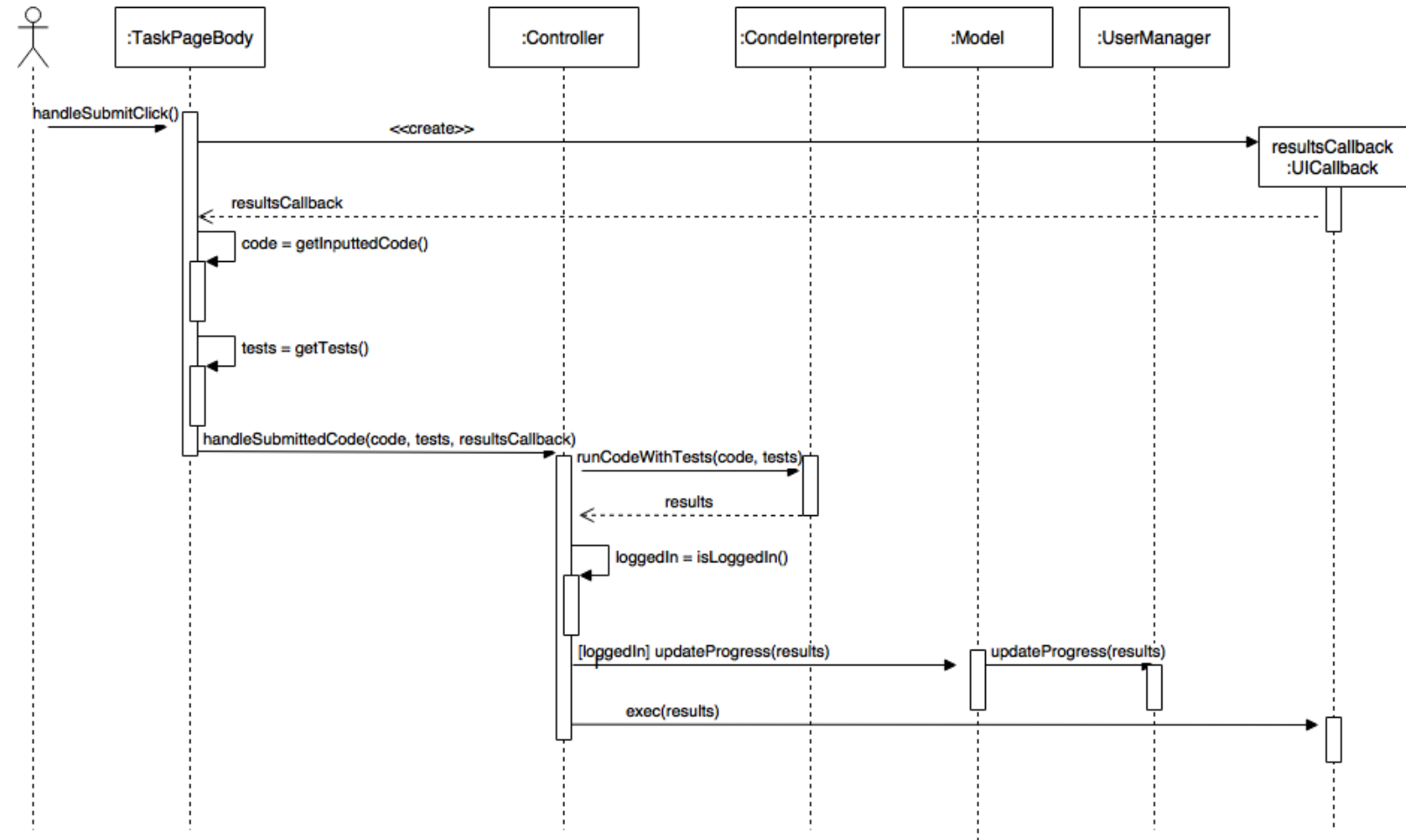


Ian Char

Requirement IDs: U-02, F-01

Use Case IDs: UC-02

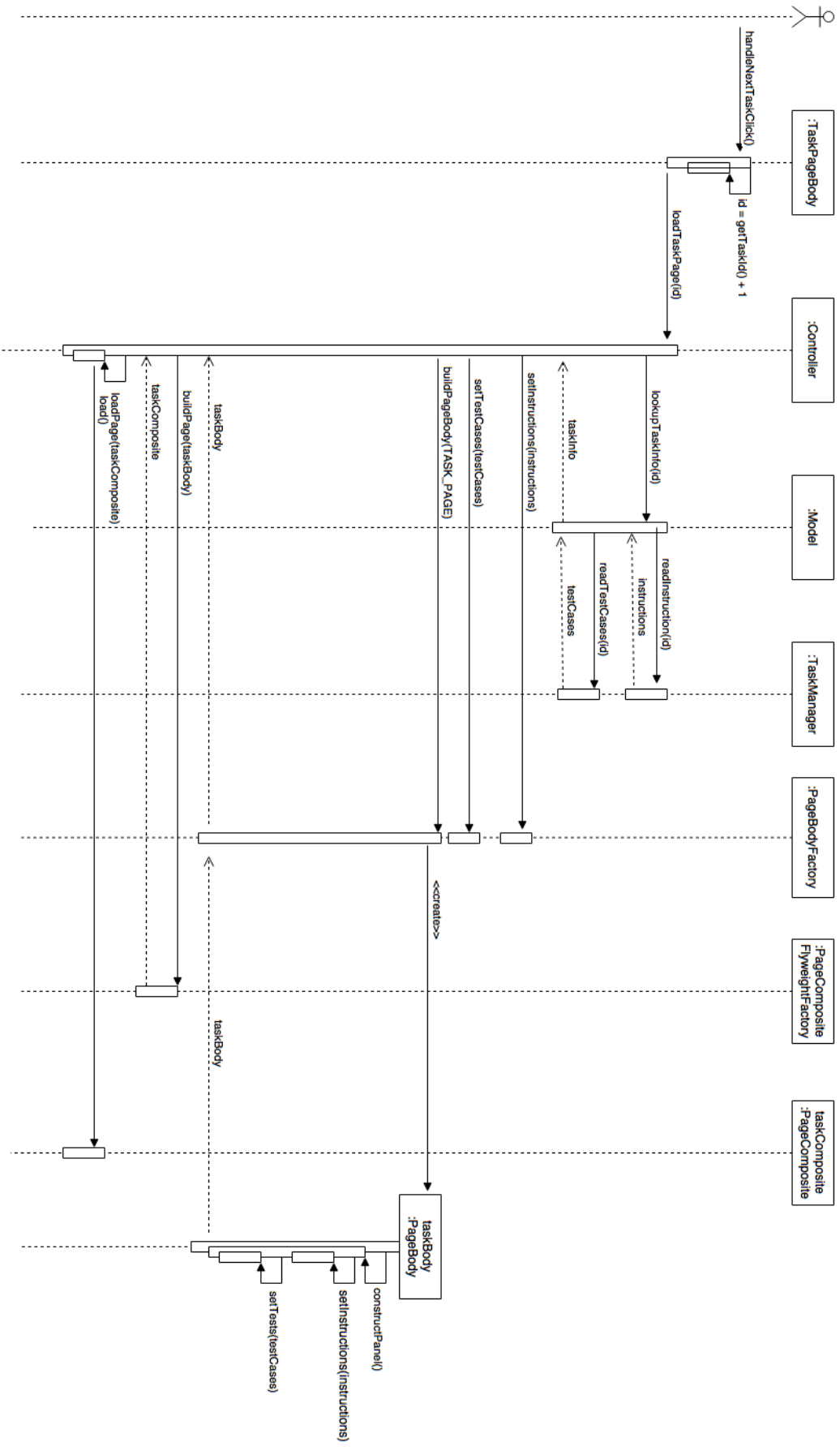
Description: This first sequence diagram shows what happens when the user submits code up to when the "continue to next task" prompt is shown..
We assume that there is no error in the code and that all the test cases were passed.



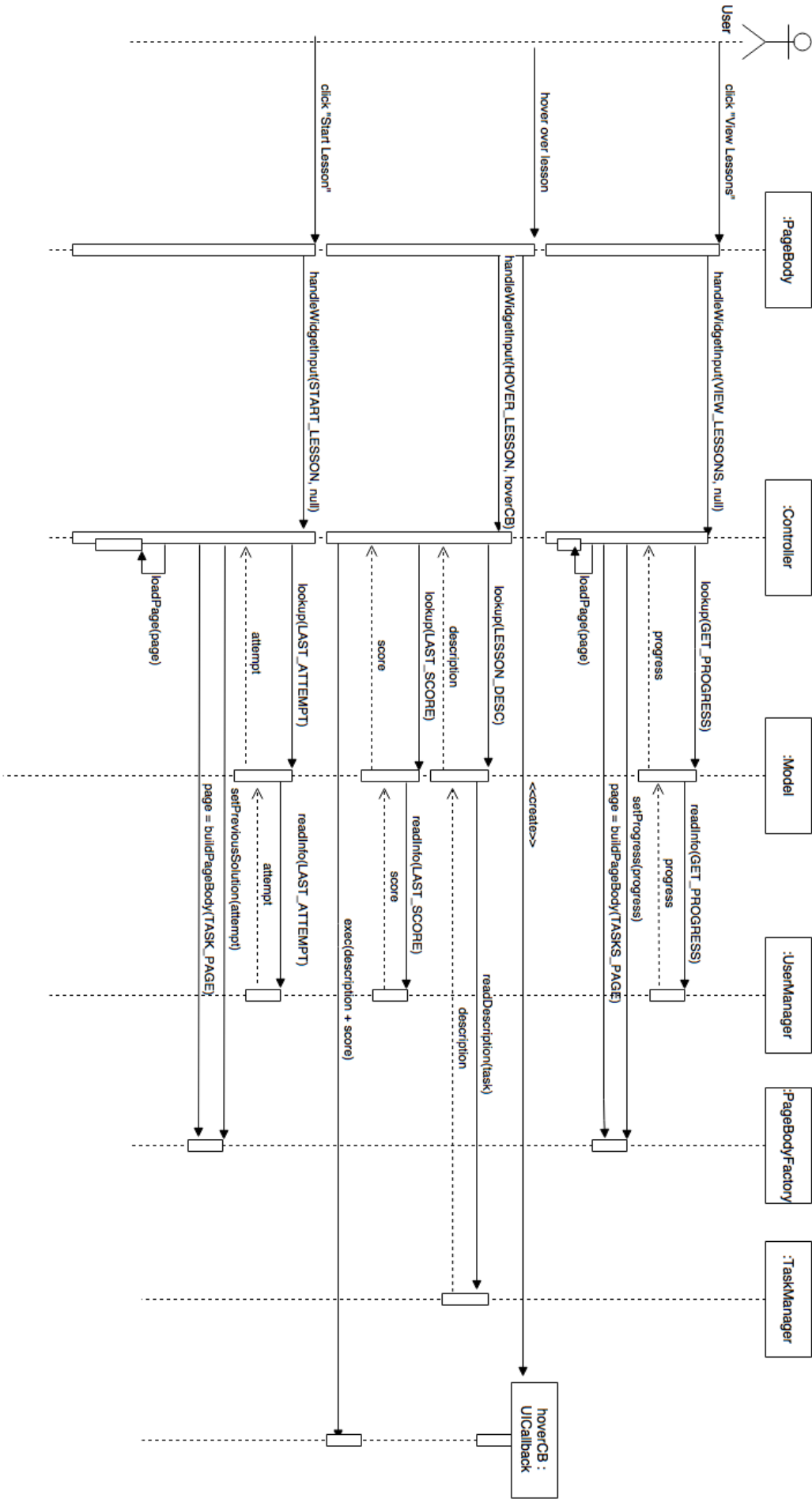
Ian Char

Requirement IDs: U-08
Use Case IDs: UC-08

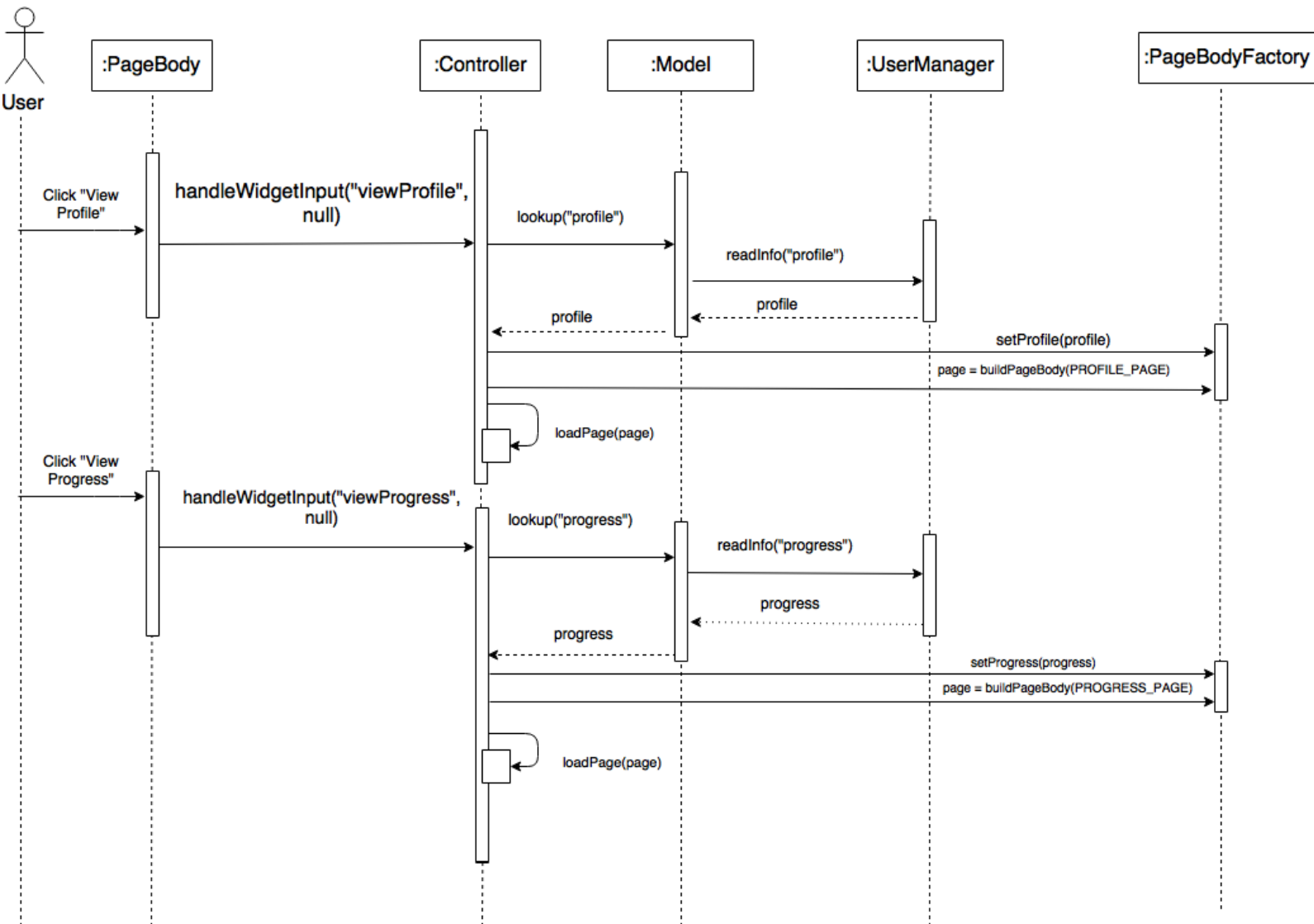
Description: This first sequence diagram shows what happens when the user submits code up to when the "continue to next task" prompt is shown.. We assume that there is no error in the code and that all the test cases were passed.

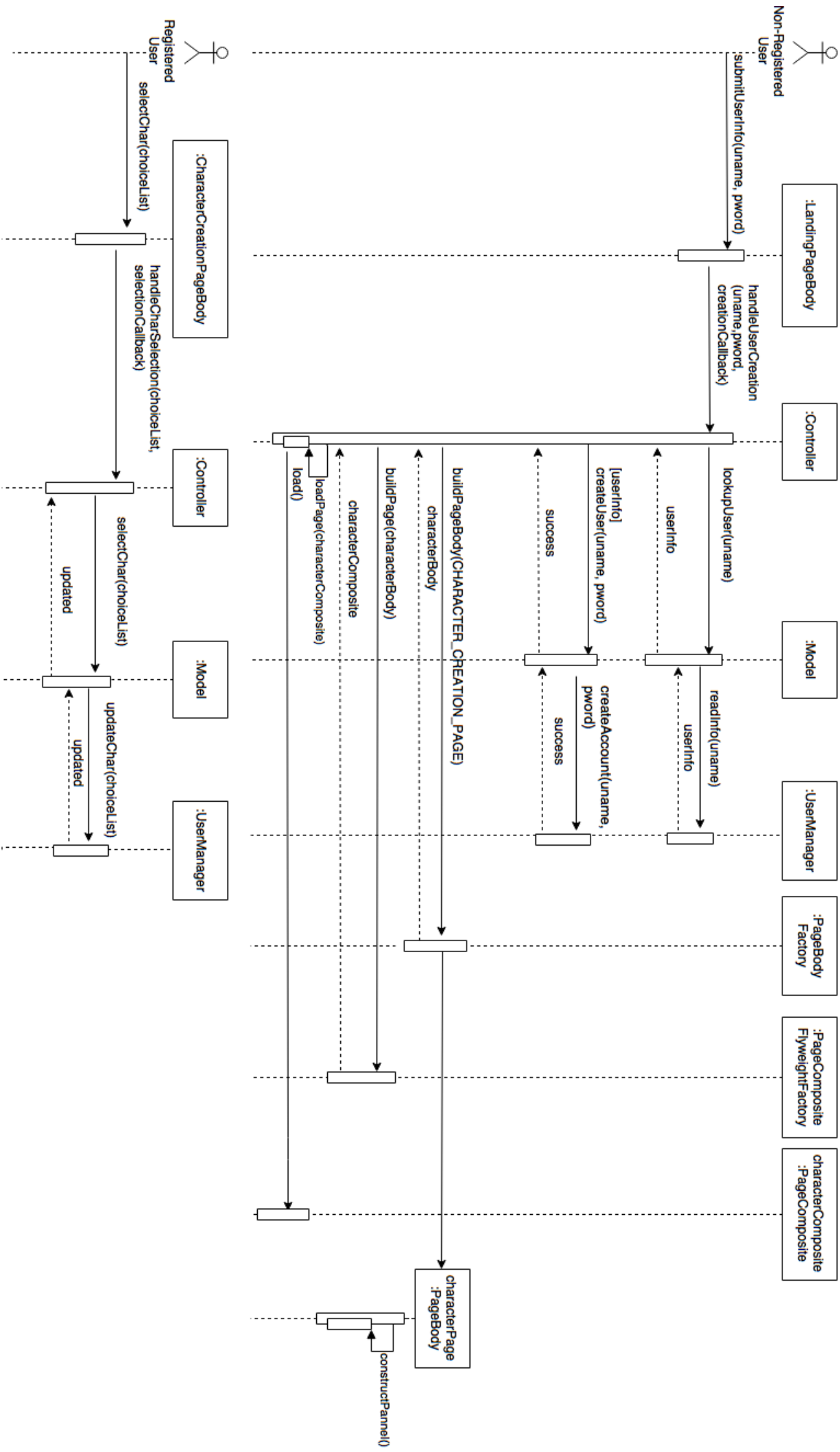


Alan Moy
Requirement 03
Use Case 03
View available lessons and navigate to one



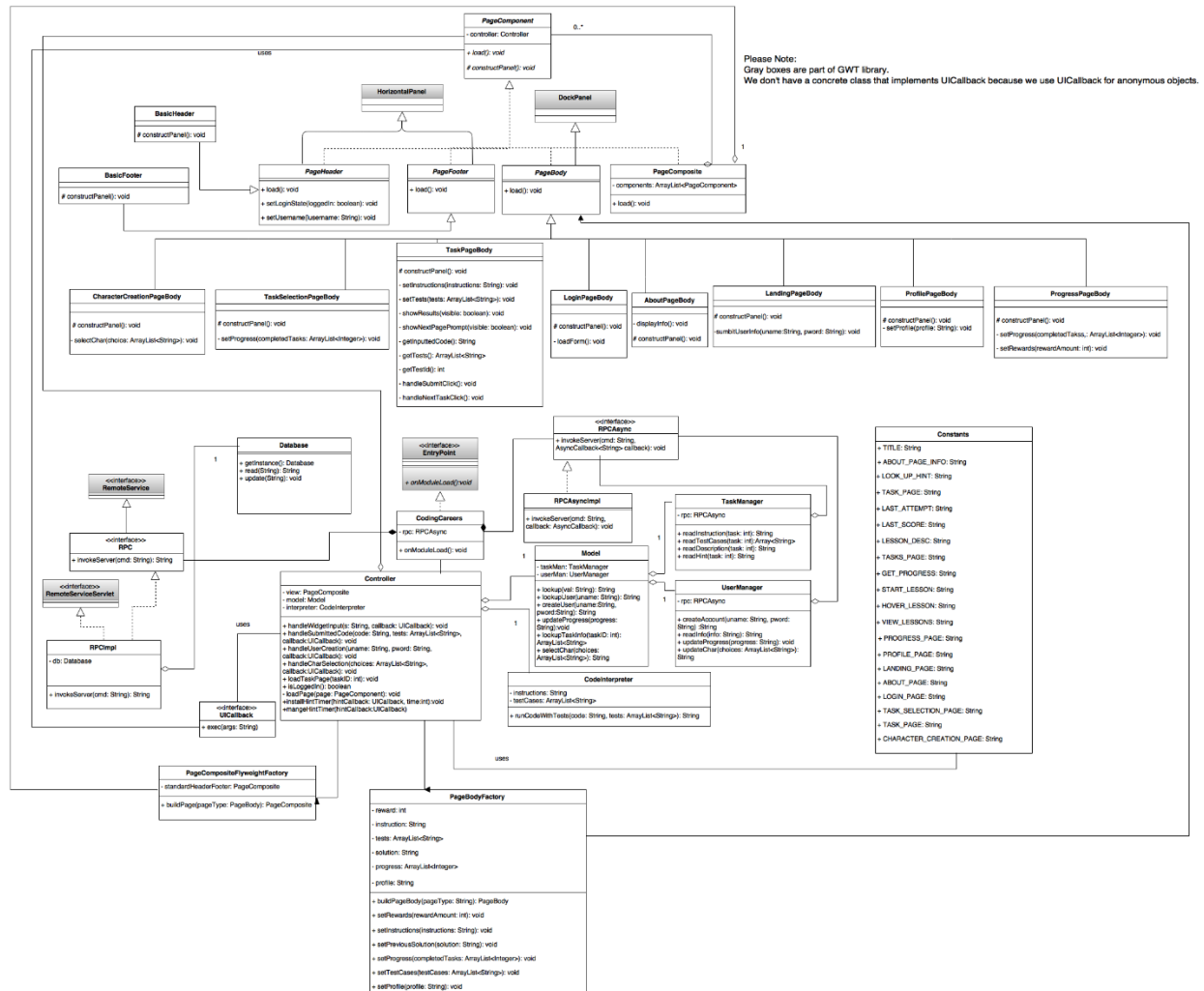
Name: Peilun Zhang
Case Name:View Progress
Description: User could view his progress of lessons
Requirement ID: 12
Use Case ID: 12

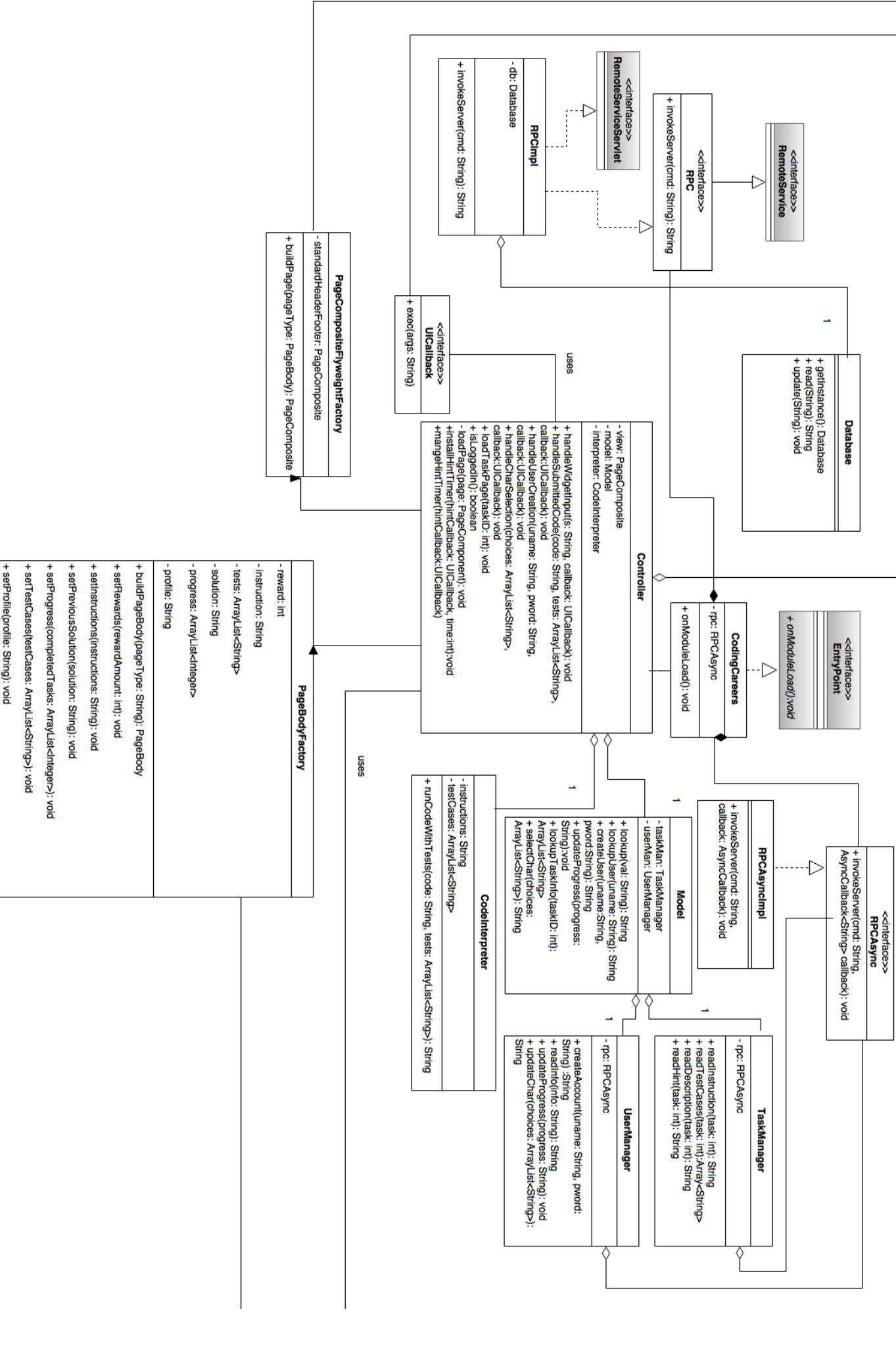


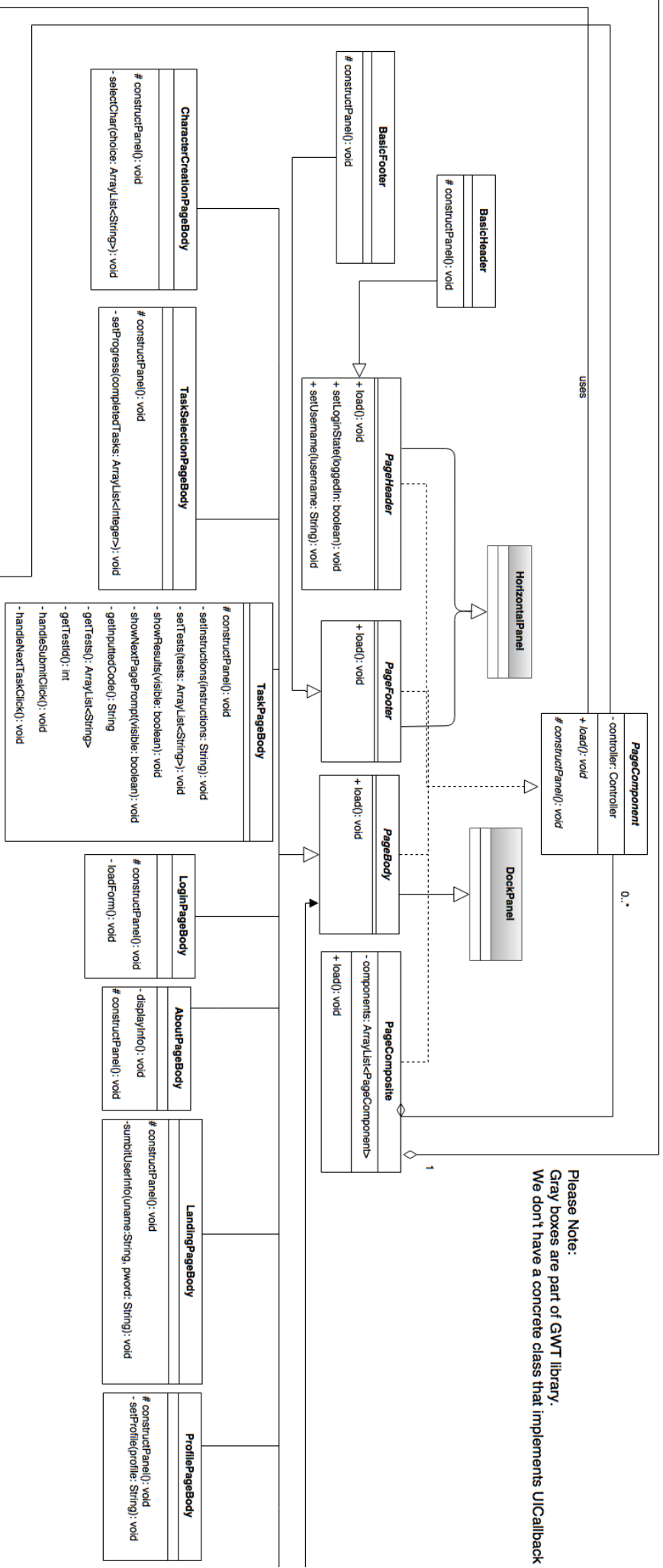


Class Diagram (See zoomed in parts on following pages):

To see a larger version go to: <https://drive.google.com/file/d/0B-WAj1ufusdOMXA2Mkh3QjRhajQ/view?usp=sharing>

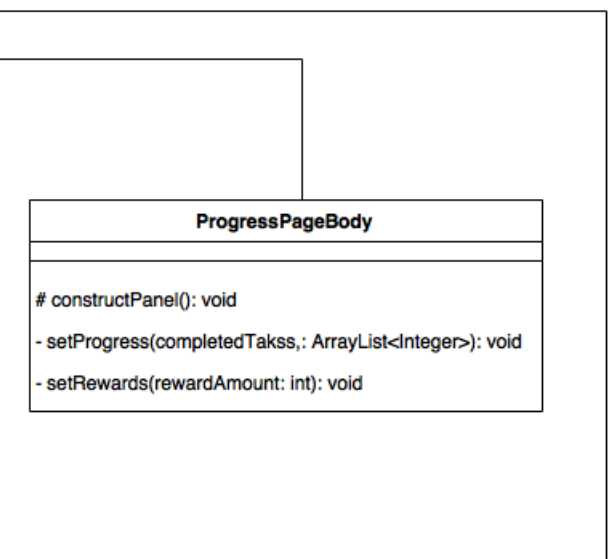






Please Note:
Gray boxes are part of GWT library.
We don't have a concrete class that implements `UICallback`

c because we use `UICallback` for anonymous objects.



Constants
+ TITLE: String
+ ABOUT_PAGE_INFO: String
+ LOOK_UP_HINT: String
+ TASK_PAGE: String
+ LAST_ATTEMPT: String
+ LAST_SCORE: String
+ LESSON_DESC: String
+ TASKS_PAGE: String
+ GET_PROGRESS: String
+ START_LESSON: String
+ HOVER_LESSON: String
+ VIEW_LESSONS: String
+ PROGRESS_PAGE: String
+ PROFILE_PAGE: String
+ LANDING_PAGE: String
+ ABOUT_PAGE: String
+ LOGIN_PAGE: String
+ TASK_SELECTION_PAGE: String
+ TASK_PAGE: String
+ CHARACTER_CREATION_PAGE: String