

**CS5760 - Human Computer Interaction & Usability Testing**

Spring 2023

# **Programming Analogies**

## **User Goals Document**

**Application name: Programming Analogies**

UX Consultants:

Ketan Patil  
Kirk Thelen

Development Team:

Grayson Wagner - Product Owner  
Emilie Rummer - Technical Lead  
Dee Paulson - Developer  
Ethan Jones - Developer  
Jack Grant - Developer  
Josh Staples - Developer  
Kevin Kulich - Developer

# User Goals Document

## Application

- **Name:** *Programming Analogies*
- **Description:**

Programming Analogies is an application that can assist instructors in better connecting programming concepts with real-world scenarios or experiences in order to enhance students' understanding and help them grasp the underlying knowledge easily. This app enables instructional staff members (teachers, teaching assistants, and professors) in quickly capturing structured analogies that may assist them in better explaining and conveying programming concepts to their students, thereby expanding the options for understandable and accessible programming education.

## Users

<b>Types of users</b>	<b>Description</b>	<b>Role</b>	<b>Expected age range</b>	<b>Expertise with technology</b>
Computer Science Instructors/ Professors	Instructors from the Computer Science department of the university/school.	Computer Science instructors will use the application to teach programming concepts to students using analogies stored in the database. Instructors would also be able to add, edit, and update analogies.	28-64	Intermediate to Expert
Undergrad Teaching Assistants	Senior undergraduate students who are Teaching Assistants for Instructors and Professors.	Undergraduate Teaching Assistants will use the application to teach programming concepts to students using analogies stored in the database. TA's would also be able to add, edit, and update analogies.	20-24	Novice to Advanced
Graduate Teaching Assistants	Graduate students who are Teaching Assistants for	Graduate Teaching Assistants will use the application to teach	23-36	Intermediate to Expert

	Instructors and Professors.	programming concepts to students using analogies stored in the database. TA's would also be able to add, edit, and update analogies.		
Students	Students of the university/school.	Students will be learning programming concepts through analogies taught by Computer Science Instructors/Professors, Undergraduate and Graduate Teaching Assistants of the university/school.	16-28	Novice

User Goals

Users	Goals
Computer Science Instructors/Professors	<ul style="list-style-type: none"> <li>● <b>Effective</b> and <b>Utility</b>: Computer Science Instructors/Professors primary goal would be to get comfortable with all the features of the application in order to fully utilize it.</li> <li>● <b>Memorable</b>: App features and functionalities should be easy to remember.</li> <li>● <b>Learnable</b>: App should be easy to learn and comprehend.</li> <li>● <b>Ergonomics</b>: Instructors/Professors should be able to use the app in a variety of settings because they are continuously bombarded with emails, messages from students, etc and basic work responsibilities.</li> <li>● <b>Effective</b>: It is important to ensure that students understand the concepts through the analogies used by the Instructors/Professors.</li> </ul>
Undergrad Teaching Assistants	<ul style="list-style-type: none"> <li>● <b>Learnable</b>: App should be easy to learn and comprehend.</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Memorable:</b> App features and functionalities should be easy to remember.</li> </ul>
Graduate Teaching Assistants	<ul style="list-style-type: none"> <li>● <b>Utility:</b> Graduate TA's primary goal would be to get comfortable with all the features of the application in order to fully utilize it.</li> <li>● <b>Learnable:</b> App should be easy to learn and comprehend.</li> <li>● <b>Effective:</b> It is important to ensure that students understand the concepts through the analogies used by the Graduate TA's.</li> </ul>
Students	<ul style="list-style-type: none"> <li>● <b>Learnable:</b> App should be easy to learn and comprehend.</li> <li>● <b>Utility:</b> App should have all the necessary features and functionalities.</li> <li>● <b>Memorable:</b> App features and functionalities should be easy to remember.</li> <li>● <b>Accessible:</b> App should be easy to use for students with disabilities.</li> <li>● <b>Ergonomics:</b> Students should be able to use the app in a variety of settings.</li> </ul>