GarnetCode

Coffee Coders

Jack Atkinson
Joshua Coleman
Cole Kennedy
Lauren Gladney
Christopher Tytone

Software Requirements Specification
Version 1.0
January 25, 2026

Table of Contents

1. Introduction

Purpose

The University of South Carolina had roughly 4,173 students enrolled in the Molinaroli College of Engineering and Computing during the 2025-2026 academic year, which accounted for more than a tenth of the school's total enrollment. This number of students is both large, and constantly increasing, with enrollment numbers consistently rising each year. This has invariably led to limitations on receiving personalized help. This is why we're striving to create a platform where these students can receive help for any of the classes they are currently taking. Our website will offer coding problems and solutions specific to the classes offered at the University of South Carolina, and also provide interviews to help prepare students beyond the academic setting.

Source:
https://sc.edu/about/offices_and_divisions/institutional_research_assessment_and_analytics/uofsc_data_dashboards/enrollment/overview.php

Scope

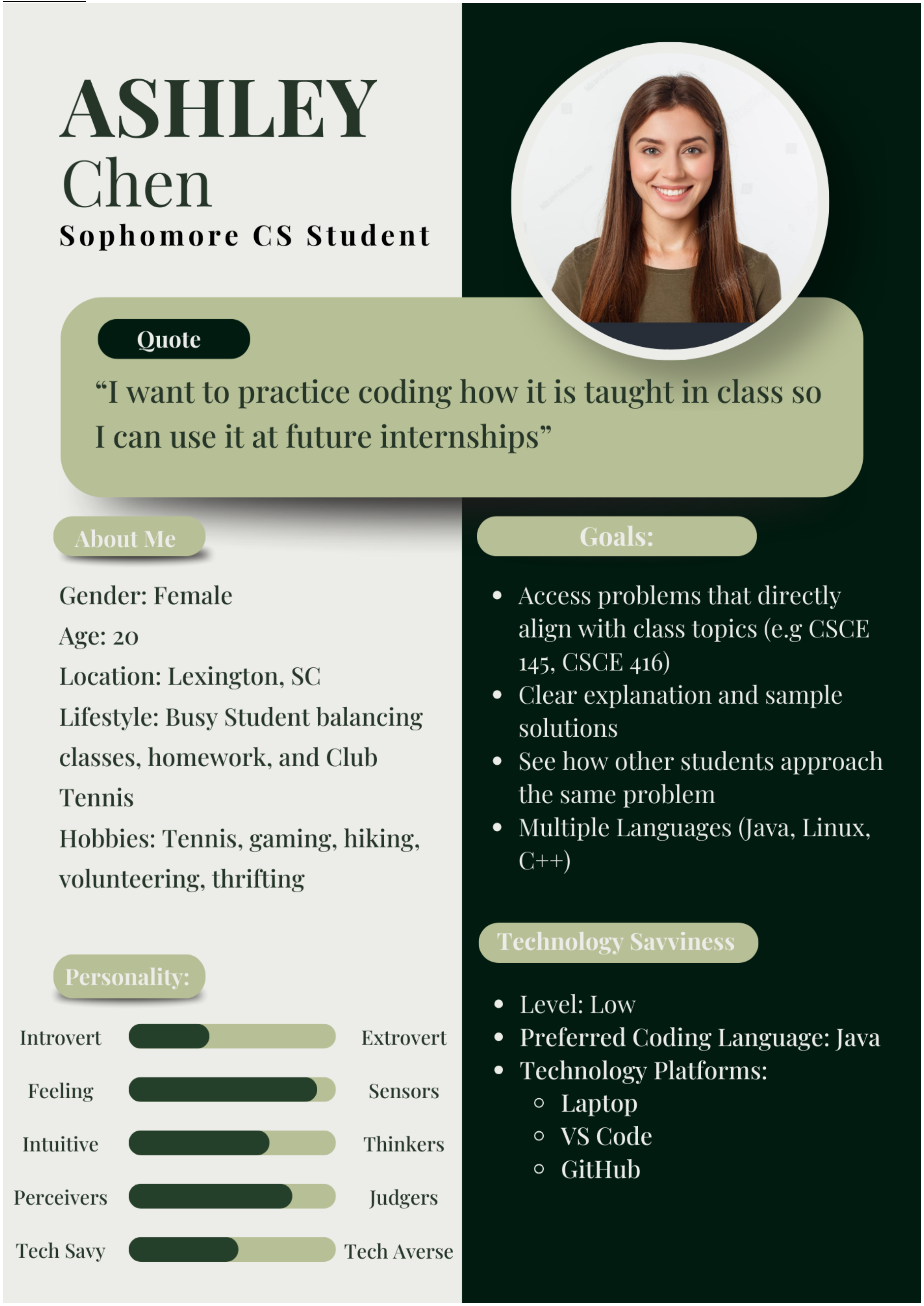
This document will overview:

- The personas of potential users and stakeholders invested in this project
- Applicable constraints to this project.
- A description of the app, along with its business use cases.
- A display of the functional and non-functional requirements of the app.
- A competitive analysis to demonstrate the purpose of the project.

2. Stakeholders

- University of South Carolina's Students
  - Undergraduate
  - Graduate
- University of South Carolina's Administrator's
  - University Administration/Leadership
  - Department Chairs
- Professors & TAs
- Companies
  - Recruiters
- Hobbyist Coders

Personas



ASHLEY Chen

Sophomore CS Student

Quote

"I want to practice coding how it is taught in class so I can use it at future internships"

About Me

Gender: Female

Age: 20

Location: Lexington, SC

Lifestyle: Busy Student balancing classes, homework, and Club Tennis

Hobbies: Tennis, gaming, hiking, volunteering, thrifting

Personality:

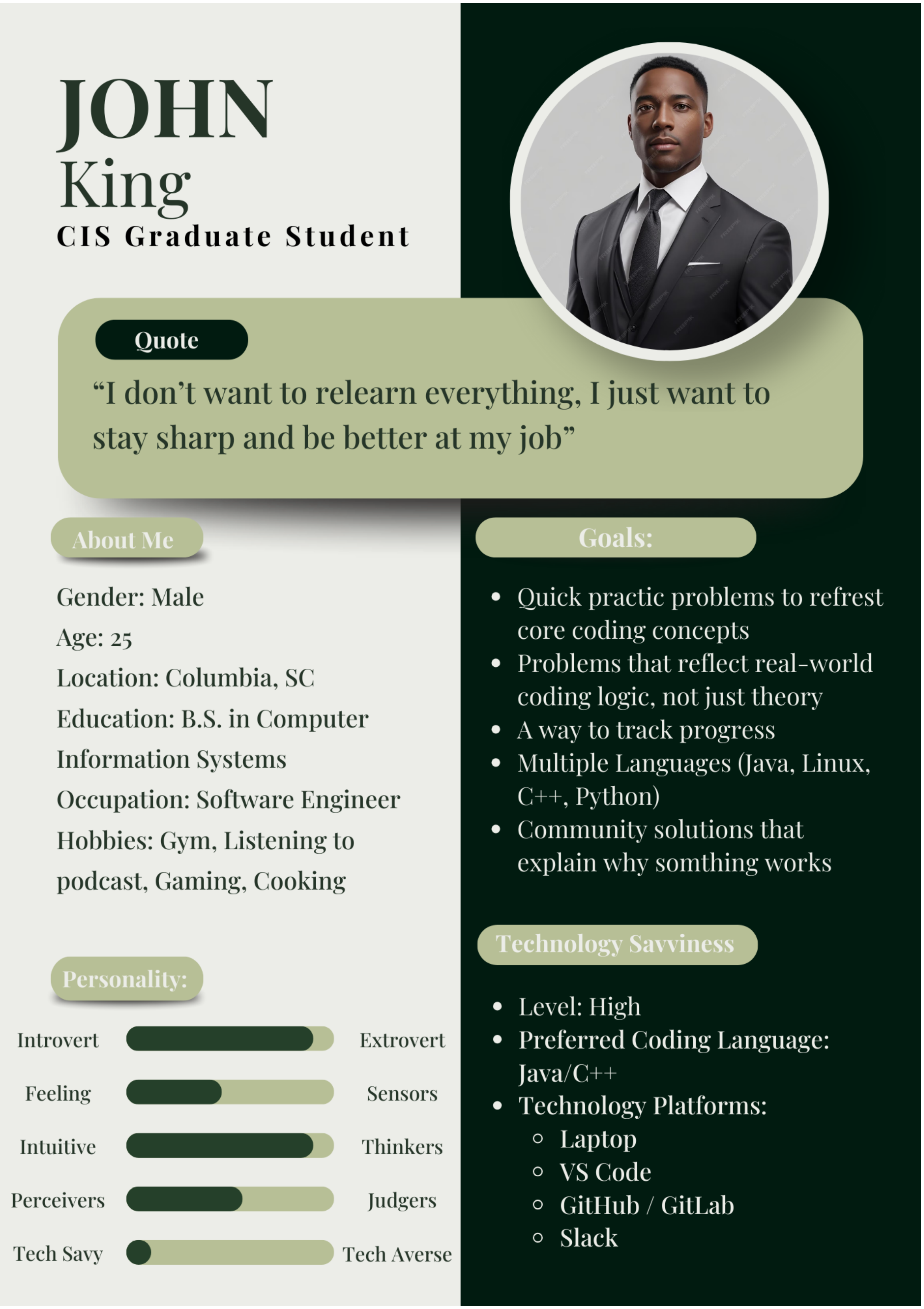
Introvert	Extrovert
Feeling	Sensors
Intuitive	Thinkers
Perceivers	Judgers
Tech Savy	Tech Averse

Goals:

- Access problems that directly align with class topics (e.g CSCE 145, CSCE 416)
- Clear explanation and sample solutions
- See how other students approach the same problem
- Multiple Languages (Java, Linux, C++)

Technology Savviness

- Level: Low
- Preferred Coding Language: Java
- Technology Platforms:
  - Laptop
  - VS Code
  - GitHub



JOHN King

CIS Graduate Student

Quote

"I don't want to relearn everything, I just want to stay sharp and be better at my job"

About Me

Gender: Male

Age: 25

Location: Columbia, SC

Education: B.S. in Computer Information Systems

Occupation: Software Engineer

Hobbies: Gym, Listening to podcast, Gaming, Cooking

Personality:

Introvert — Extrovert

Feeling — Sensors

Intuitive — Thinkers

Perceivers — Judgers

Tech Savy — Tech Averse

Goals:

- Quick practic problems to refrest core coding concepts
- Problems that reflect real-world coding logic, not just theory
- A way to track progress
- Multiple Languages (Java, Linux, C++, Python)
- Community solutions that explain why somthing works

Technology Savviness

- Level: High
- Preferred Coding Language: Java/C++
- Technology Platforms:
  - Laptop
  - VS Code
  - GitHub / GitLab
  - Slack

DR. SAM Carter

Professor

Quote

"I want students to get real-world application practice in a way that aligns with what we teach"

About Me

Gender: Female

Age: 45

Location: Columbia, SC

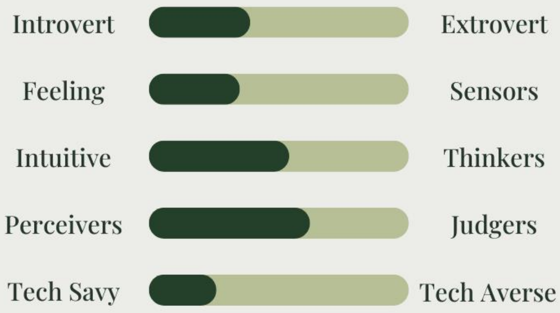
Education: PhD in Computer Science

Hobbies: Reading research papers, golfing, painting

Goals:

- A platform where students can practice coding concepts taught in lecture
- Course-aligned problem sets that reinforce weekly topics
- Commenting and discussion to correct misconceptions

Personality:



Technology Savviness

- Level: High
- Preferred Coding Language: C++ or Java
- Technology Platforms:
  - Desktop
  - VS Code / IntelliJ
  - GitHub

3. Constraints

Time Constraints

- This project must be completed within the academic semester, according to the course schedule.

Monetary Constraints

- This project has an allocated development budget of $0.

Technical Constraints

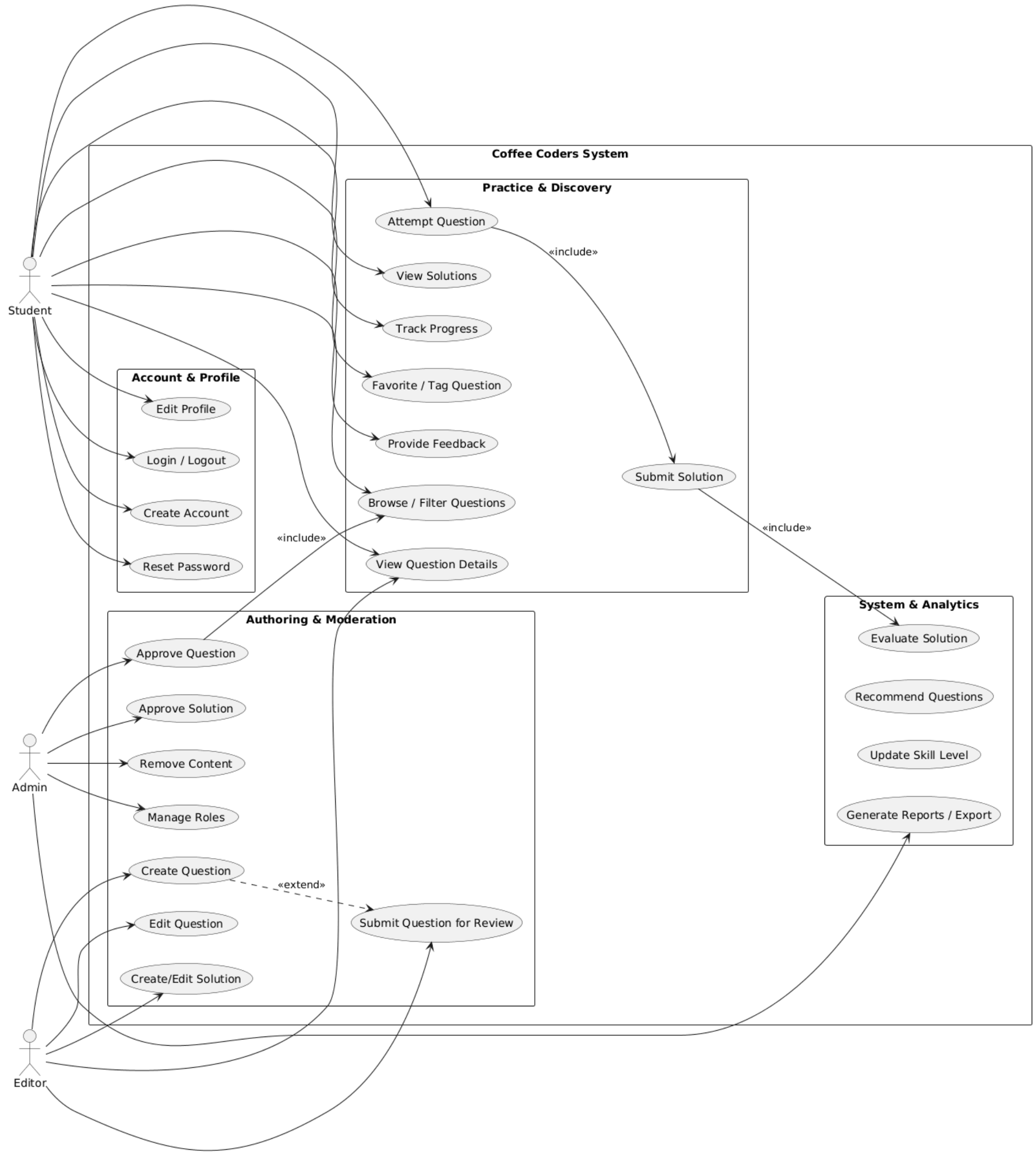
- The system must be accessible through standard web browsers on commonly used devices, including desktops, laptops, and mobile devices.
- The code must be done in Java.

4. Overall Description

This app must be accessible to University of South Carolina students and targets computer science and engineering majors. Users will be able to access the app through their web browsers on laptops, desktops, and mobile devices. The app will have a collection of coding and concept-based practice problems tailored to USC courses and interview preparation. Users will be able to interact with these problems by submitting solutions, viewing explanations, tracking their progress, and being able to comment with others and provide feedback.

5. Business Use Cases

Include a Use Case diagram, depicting the high-level goals of your system.



6. Functional Requirements

[Functional Requirements]

7. Non-Functional Requirements

For each section below, list any relevant non-functional requirements.

• Look and feel requirements

- The system shall provide a clean, organized interface that emphasizes coding problems and solutions.
- The system shall be visually usable on desktops, laptops, and mobile devices.
- The system shall clearly provide elements such as buttons, links, and input fields.

• Usability requirements

- The system shall provide clear error messages when user input is invalid.
- The system shall allow users to navigate between questions, solutions, and progress

• Performance requirements

- The system shall load pages quickly during normal use.
- The system shall show filtered results without a noticeable delay.
- The system shall support multiple users at the same time.
- The system shall continue working during busy times such as exams and finals.

• Maintainability and support requirements

- The system shall be written in a way that makes it easy to add new features later.
- The system shall include basic documentation for future developers.
- The system shall allow administrators to update or remove content without shutting down the system.
- The system shall record system errors to help with fixing problems.

• Security requirements

- The system shall require users to log in before changing data.
- The system shall limit admin features to authorized users.
- The system shall protect user login information.
- The system shall check user input to prevent security issues.
- The system shall support secure login for USC users.

• Cultural requirements

- The system shall use respectful and inclusive language.
- The system shall avoid content that could be offensive.
- The system shall be usable by people with different skill levels.

• Legal Requirements

- The system shall follow university rules for academic integrity.
- The system shall protect user privacy.
- The system shall require users to agree to terms of use when creating an account.

8. Definitions and Acronyms

GarnetCode: The name of the software

User: Any person who interacts with the system

Student: A user enrolled at USC who uses the system, signed in using USC CAS

Editor: A user who is allowed to create and edit questions and solutions

Administrator (Admin): A user who manages the system and approves questions and solutions

Guest Mode: A way to use the system without creating an account. Progress isnt saved in this mode.

Question: A coding or concept-based problem presented to users for practice.

Solution: A correct answer or explanation for a question.

Hint: A short message that helps guide the user towards the correct solution

Tag: A label used to organize questions by topic, course, difficulty, or language

9. Competitive Analysis

Features	**LeetCode**	**HackerRank**	**AlgoExpert**
Coding practice problems	Yes	Yes	Yes
Problem Library	1,000s of algorithm & interview problems	Many problems + multiple domains	Smaller set (~160-220) with focus on interview patterns
User Discussion / Comments	Yes – limited to discussion threads	Yes – Community challenges & discussion	No – generated by problems
User can post solutions	Yes	Yes	No
Interview Prep Focus	Yes (Company-tailored questions)	Yes	Yes (generalized interview prep)
Mock Interview Tool	Yes (Premium)	No	Yes
Company-Style Interview Questions	Yes (Premium)	Limited	No
Free vs. Paid	Free but premium feature ($35/mo)	Free	$99/yr
Leaning Resources	Tutorials, learning cards, community solutions	Tutorials, domain learning pathways	Video explanations & in-depth walkthroughs

LeetCode, HackerRank, and AlgoExpert all represent three different approaches to technical interview and coding practice. LeetCode is ideal for thorough interview preparation because it provides the largest library of algorithm and data structure problems with user discussions and company-tailored questions. HackerRank, while not as specialized for interviews, offers a variety of coding challenges across multiple domains and is beneficial for beginners to intermediate practice. Though it has fewer problems and less community involvement than LeetCode, AlgoExpert focuses on carefully chosen problems with video explanations and structured learning. Since none of these platforms fully support student-generated problems with integrated comments and collaborative solutions. This gap allows our app to be tailored specifically for engineering students at the University of South Carolina by incorporating course-aligned practice problems, learning progress tracking, and peer posting features that allow posts from teachers, teaching assistants, and students alike.

10. References

https://sc.edu/about/offices_and_divisions/institutional_research_assessment_and_analytics/uofsc_data_dashboards/enrollment/overview.php

11. Appendices

Any relevant information that can assist in understanding the requirements.