**CSC 3150 – System Specification**

**(Design Document)**

**Be sure to submit this artifact to Canvas by the due date.**

**NOTE: No assignment submissions can be accepted**

**after the last day of class. No exceptions.**

# **Cover Page**

.

# 時間の検索結果 | かわいいフリー素材集 いらすとや

# **CLace is a time-keeping app based on geography that offers the best help to SPU students who are usually late for class time. In the series of documents, we covered how CLace came up, how it can benefit our stakeholders, and how we will build CLace and achieve the goal most efficiently. This document will show you the dipper investigation with much more details, functions, data, plans, and designs to announce CLace to our development team.**

# **Table of Contents**

Table of Contents

[**Cover Page** 1](#_Toc168595460)

[**Table of Contents** 1](#_Toc168595461)

[**1.** **Executive Summary** 1](#_Toc168595462)

[**2.** **Introduction** 2](#_Toc168595463)

[**2.1.** **Problem Statement / Project Vision** 2](#_Toc168595464)

[**2.2.** **System Capabilities** 2](#_Toc168595465)

[**2.3.** **Non-functional Requirements and Design Constraints** 2](#_Toc168595466)

[**2.4.** **System Evolution** 2](#_Toc168595467)

[**2.5.** **Document Outline** 3](#_Toc168595468)

[**3.** **Structural Model** 3](#_Toc168595469)

[**3.1.** **Model Introduction** 3](#_Toc168595470)

[**3.2.** **Class Diagrams** 4](#_Toc168595471)

[**3.3.** **Metadata** 5](#_Toc168595472)

[**4.** **Architecture Design** 13](#_Toc168595477)

[**4.1.** **Architecture Overview** 13](#_Toc168595478)

[**4.2.** **Infrastructure Model** 13](#_Toc168595479)

[**4.2.1.** **Deployment Diagram 1 – Architecture Overview** 13](#_Toc168595480)

[**4.2.2.** **Deployment Diagram 2 – Nodes and Artifacts** 14](#_Toc168595481)

[**4.3.** **Hardware and Software Requirements** 14](#_Toc168595482)

[**4.3.1.** **Hardware Components** 14](#_Toc168595483)

[**4.3.2.** **Required Software Components** 14](#_Toc168595484)

[**4.4.** **Security Plan** 15](#_Toc168595485)

[**4.4.1.** **Security Overview** 15](#_Toc168595486)

[**4.4.2.** **Security Plan** 15](#_Toc168595487)

[**15.** **User-Interface** 16](#_Toc168595488)

[**a.** **User-Interface Requirements and Constraints** 16](#_Toc168595489)

[**b.** **Window/Screen Navigation Diagram** 16](#_Toc168595490)

[**c.** **UI Wireframes** 17](#_Toc168595491)

[**d.** **Reports: "Formal Output" Design** 17](#_Toc168595492)

[**16.** **Appendices** 17](#_Toc168595493)

[**a.** **Glossary** 17](#_Toc168595494)

[b. **References / Bibliography** 17](#_Toc168595495)

[**c.** **Supporting documentation** 18](#_Toc168595496)

# **Executive Summary**

This document serves as the system specification for the CLace development team. CLace is a timekeeping application designed for Seattle Pacific University (SPU) students to help them improve their punctuality and time management skills. The primary audience for this document includes developers, project managers, and stakeholders involved in the development and implementation of CLace. The project request is approved, has a system proposal, and goes to the system specification step to be in the world. It will include the structural model, architecture model, security plan, and user interface.

# **Introduction**

CLace aims to help students improve punctuality by providing a motivational timekeeping app. It records time habits, sets milestones, calculates travel time, and features a ranking dashboard to encourage timely arrivals. CLace targets students who often struggle with self-control, promoting better time management habits crucial for modern society. More information is provided in the system proposal part 2. Section 1.0.

## **Problem Statement / Project Vision**

CLace is an application that offers much more precise information about time to students who are usually late for class. We hope CLace will improve the classroom environment and students’ successful habit-building. CLace will advise students about the exact amount of time they need to spend in class and improve their time management.

CLace also benefits the other stakeholders. The teachers will not be disrupted by the late students. The student’s friends can join the journey to be on time to the class with the existing CLace user together. CLace's primary target, young students, can be reached by the advertisement sponsors. CLace will be a different timekeeping application with a geographic feature. With this unique trait, we hope CLace will be a new milestone in the new era of the alarm.

## **System Capabilities**

CLace will ring the alarm ahead of class time, which is calculated by the distance. Then, CLace will start the countdown if the students start to move.

-Use Case a2. Start the countdown.

CLace will check the successes and failures when the students arrive at the classroom.

-Use Case a3, 4. Report the success/failure.

CLace will have the ranking dashboard.

-Use Case b3. Check the group dashboard.

CLace will allow the users to set the class time and destination.

-Use Case a1. Set the class time.

*System capabilities are described in the system proposal in Section 5.0. Use-case diagram and description.*

## **Non-functional Requirements and Design Constraints**

CLace must be compatible with Android and Google Maps.

CLace should find a way to calculate the distance between the building entrance and the classroom door.

CLace should find the shortest path.

CLace has a user interface that looks like a racing game.

CLace should work in every Seattle Pacific University area and building.

CLace has a safe environment in terms of security.

CLace allows further updates after launching.

CLace has direct symbols to let users know what to do to operate the system.

## **System Evolution**

CLace will improve the map function and social function for the next version.

* + 1. **Version 2 Changes**

CLace will collect the group members' scores in the group dashboard and then hold a weekly competition between some groups.

CLace will calculate the user’s speed and update the required time to the classroom simultaneously.

* + 1. **Version 3 and beyond Changes**

CLace will expand the service area further than SPU to be the time-keeping app for students and other customers with various backgrounds.

CLace will strengthen group-making and searching by providing more categories and better search algorithms.

## **Document Outline**

CLace is a geometry-based timekeeping application that prevents the tardiness of SPU students. CLace should be connected with Google Maps and offer the time to go to the class based on the distance between the student and the destination. We expect this application will lower the tardiness rate of the class and help students build healthy time habits.

Read the CLace system proposal Section 1.0 Introduction and Overview for more detailed information.

# **Structural Model**

## **Model Introduction**

This part will show you how the CLace organs will work and give you instructions on how to build the actual system for CLace. The next part will cover the UML class diagrams and their descriptions to inform you how we designed them.

## **Class Diagrams**

A computer screen shot of a computer program

Description automatically generated

## **Metadata**

Dashboard Class diagram

A screenshot of a computer

Description automatically generated

Description: Represents a dashboard that members will see and compete each other

Visibility: Public

Is Abstract: No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| groupName | Group’s name | No | 1 |
| Category | Group’s common interests | No | 1..\* |
| groupMembers | Students in the group | No | 1..\* |
| maxMember | Group dashboard’s capacity | No | 1 |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is polymorphic? |
| postGroup | Create the group and post it to be visible | Yes | No |
| getName | Outputs group name | Yes | Yes |
| getCategory | Outputs group’s interested category | Yes | Yes |
| getGroupMembers | Outputs the number of group members | Yes | No |
| getMaxMember | Outputs the group’s capacity | Yes | No |
| addMember | Add the Member to the group | No | No |

**Processing Outlines 🡪**

**postGroup()**

Save the group and upload it

**getName()**

Returns the group dashboard name in string.

**getCategory()**

Returns the list of group’s categories in string.

**getGroupMembers()**

Returns the list of group members’ names in string.

**getMaxMember()**

Returns the capacity in numbers of members for the group in integer.

**addMember(Member)**

gets Member class and adds him/her to the group dashboard.

Map Class diagram

A screenshot of a computer program

Description automatically generated

Description: Represents a map, the key feature of CLace.

Visibility: Public

Is Abstract: No

Additional Information: the time calculated by the distance/3.2mph. We use the physics formula of time distance/speed and an average human walking speed of 3.2mph.

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| name | User’s custom name of schedule | No | 1 |
| destination | User’s goal that user wants to be on time | No | 1..\* |
| classSchedule | User’s class start time | No | 0..\* |
| userLocation | User’s current location | No | 1 |
| GoogleMap | Gets the information from Google Maps | Yes | 1 |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is polymorphic? |
| getName | Outputs the schedule’s name | Yes | Yes |
| getDestination | Outputs the location of the destination to go. | Yes | No |
| getClassSchedule | Saves and Outputs the class schedule | Yes | No |
| calculateTime | Calculate the time needed to go to the destination | Yes | No |
| getGoogleMap | Reads the Google Maps information | No | No |

**Processing Outlines 🡪**

**getName()**

Returns the schedule name in string.

**getDestination()**

Returns the user’s destination in string.

**getClassSchedule()**

Returns the class time in Time.

**calculateTime(destination, userlocation)**

**if**

(getGoogleMap(destination)-getGoogleMap(userlocation)/4)>0)

Return getGoogleMap(destination)-getGoogleMap(userlocation)/4;

Else

Return 0;

**getGoogleMap()**

Communicates with the Google Maps for the geographical information.

Member Class diagram

A screenshot of a computer program

Description automatically generated

Description: Represents a member who will be seen on the dashboard and uses CLace.

Visibility: Public

Is Abstract: No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| memberName | Member’s name | No | 1 |
| category | Member’s personal interests | No | 1..\* |
| score | Member’s score calculated by the success/failure | No | 1 |
| streak | Member’s streak per day | No | 0..\* |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is polymorphic? |
| getName | Outputs member name | Yes | Yes |
| getCategory | Outputs member’s interested category | Yes | Yes |
| getScore | Outputs the member’s score | Yes | No |
| getStreak | Outputs the member’s streak | Yes | No |
| Calculate-Score | Calculate the score based on the formula | Yes | No |

**Processing Outlines 🡪**

**getName()**

Returns the member’s name in string.

**getCategory()**

Returns the list of member’s categories in string.

**getScore()**

Returns the list of group members’ names in string.

**getStreak()**

Returns the capacity in numbers of members for the group in integer.

**Calculate-Score()**

Returns the score based on the success or failure.

If (checkSuccessFailure)

++score;

Else

()

Return score;

Timer Class diagram

A screenshot of a computer program

Description automatically generated

Description: Represents a timer that will decide the user’s success and failure.

Visibility: Public

Is Abstract: No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| name | Timer’s name | No | 1 |
| successFailure | Result that the user arrived at the goal on time or not. | No | 1..\* |
| alarmTime | The time that alarm will ring | No | 1..\* |
| countdown | The time that takes to the destination. | No | 1 |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is polymorphic? |
| getName | Outputs the timer name | Yes | Yes |
| getTime | Outputs the time to leave. | Yes | Yes |
| ringAlarm(getTime()) | Rings the alarm if the time becomes the desired time | Yes | No |
| checkTimer(countdown) | Check the user’s time taken to the destination | Yes | No |
| checkSuccessFailure() | Report the result of the user’s way to go to the destination. | Yes | No |

**Processing Outlines 🡪**

**getName()**

Returns the timer’s name in string.

**getTime()**

Saves and returns the class time from the user as a time.

**ringAlarm(getTime())**

Ring the alarm if the current time matches the class schedule.

If getTime() = alarmTime

Ring the alarm bell.

**checkTimer(countdown)**

Returns the countdown that is required to go to the class.

**checkSuccessFailure()**

Report the true if the result is successful and false if the result is a failure.

If (checkTimer(countdown))>=0

Return true;

Else

Return false.

# **Architecture Design**

## **Architecture Overview**

CLace will hold a 3-tier client-server system that is composed of the device, web application server, and database server. CLace will require two more new servers to operate the system that are connected by ODBC. The servers will be protected by the firewall. The internet service provider will connect the customer’s device and the system.

## **Infrastructure Model**

* + 1. **Deployment Diagram 1 – Architecture Overview**

A diagram of a computer system

Description automatically generated

* + 1. **Deployment Diagram 2 – Nodes and Artifacts**

A diagram of a cloud server

Description automatically generated

## **Hardware and Software Requirements**

CLace is an Android-based application with a geographic feature. The hardware should be the Android devices that can connect to the Google store, have a working GPS, and the devices operated by the Android. The software should have a proper Android version, working Google Maps, and an internet connection to run CLace for the customers.

* + 1. **Hardware Components**

Any Android mobile device with GPS can be used for CLace. The server will be needed, too.

* + 1. **Required Software Components**

The Android should be higher than Android 13 and compatible with API, Google Maps, and Android Studio. The development team needs the Android Studio and server to store the data. CLace also requires a firewall and external cloud server to reduce the damage caused by the accidents.

## **Security Plan**

* + 1. **Security Overview**

This section will predict the security threats from people and the environment and CLace’s plan to minimize the damage caused by unpredictable attacks. They will be man-made disasters, natural disasters, internal intruders, and external hacker attacks. The security plan will show the solutions for each disaster.

* + 1. **Security Plan**

Man-made disasters: fire, short circuit, server reset, connection loss, liquid, etc.

Natural disasters: fluids, tornadoes, earthquakes, solar winds, heavy snow, etc.

Internal intruder: Industrial spy, sabotage, terror, etc..

External hacker attack: ransomware, database spill, DDoS, malfunction, etc.

A table with numbers and text

Description automatically generated

Controls.

1. Insurance for businesses.
2. The newest firewall.
3. Security authorization.
4. Back-up device.
5. Back-up cloud server from a big company.
6. Disaster insurance.
7. Fire control drill.
8. Keep the environment clean.
9. Distinguish work and personal devices.
10. Prepare the emergency power.
11. Work-ethics.
12. The newest vaccine program.
13. Strong log-in system.
14. Cable covers.

# **User-Interface**

## **User-Interface Requirements and Constraints**

The following two images will show window/screen navigation diagrams to visualize our window/screen and UI requirements for CLace. Window/screen diagrams will help you understand the flow of the windows and how they are interconnected. UI requirement diagram will be your blueprint showing layouts, placement of the elements, and detailed design.

## **Window/Screen Navigation Diagram**

A diagram of a computer program

Description automatically generated

## **UI Wireframes**

CLace UI Wireframes.

A diagram of a map

Description automatically generated

## **Reports: "Formal Output" Design**

CLace doesn’t have formal output.

# **Appendices**

## **Glossary**

CLace – Class + Race.

ODBC - Open Database Connectivity.

TCP - Transmission Control Protocol.

IP - Internet Protocol.

HCI- Human-Computer Interaction.

## **References / Bibliography**

Cronkleton, E. (n.d.). *Average walking speed: Pace, and comparisons by age and sex*. Healthline. <https://www.healthline.com/health/exercise-fitness/average-walking-speed>

More references/bibliography can be found in the System Proposal references/bibliography.

## **Supporting documentation**

No supporting documents.