Siesta Garden Controller

Software Architecture Design

SAD Version 2.0

Team T06
11 May 2021

Ashley Krattiger
Justin Lusby
Daniel Sherwood
Matthew Zamora

CS 460 Software Engineering

Table of Contents

1 Introduction

This document has been specially prepared for use by the project's development teams and managers. The Software Architecture Design (hereafter abbreviated to SAD) document is a describes the project's architecture, provides a logical view of its components, and explores some sample use-cases for its various users. Additionally, it outlines the design constraints that impose limitations on the Siesta Gardens Controller (SGC) system. The purpose of this document is to provide the developers an outline for the logical and programmatic flow of the project.

1.1 Purpose

The purpose of this document is to outline the architecture of the SGC system. It includes several use-case diagrams that describe the interaction between the various users (including visiting guests as well as park administrators) and the components within in the system. The class diagram will provide the developers with an overview of classes, methods, and object-to-object interactions as necessitated by required functionality of the system.

1.2 Scope

The scope of the SAD is to depict the architecture of the SGC system. Users who require a technical understanding of the SGC system are encouraged to start by reading this document and then

This document is regularly updated—if changes are made during the process of implementation, then they will be reflected here; likewise, any change that occurs in this document will also affect the implementation. The developer should refer to the use-case and class diagrams as a foundation to implement the SGC system (i.e., if a method is described in one of the diagrams, it should be thus present in the implementation).

1.3 Definitions, Acronyms, & Abbreviations

The following acronyms and abbreviations will be used regularly throughout the document and are defined here for convenience:

GUI	Graphical User Interface
JDK	Java Development Kit
OMT	Object Modelling Technique
RFID	Radio Frequency Identification
SAD	Software Architecture Design
SGC	Siesta Gardens Controller
SRS	Software Requirements Specification
UI	User Interface
UUID	Universally Unique Identifier

2 Design Overview

Following is a class diagram depicting the interactions between the subsystems in the Siesta Gardens Controller. System initialization and driving is performed by the Central

Management System class, which instantiates references to the other important subsystems (Vehicle Management, Token Management, Alarm Management). Each of these subsystems is responsible for their own various functionalities, and, in general, will end up receiving and reporting information to and from the Central Management System.

The below diagram is meant as a general overview for the structure of the Siesta Gardens Controller program, and, as such, is not an exhaustive specification of the classes that comprise the program, nor of the functions within those classes. It is therefore limited to providing a general map of the dependencies between classes, and how information might travel between them.

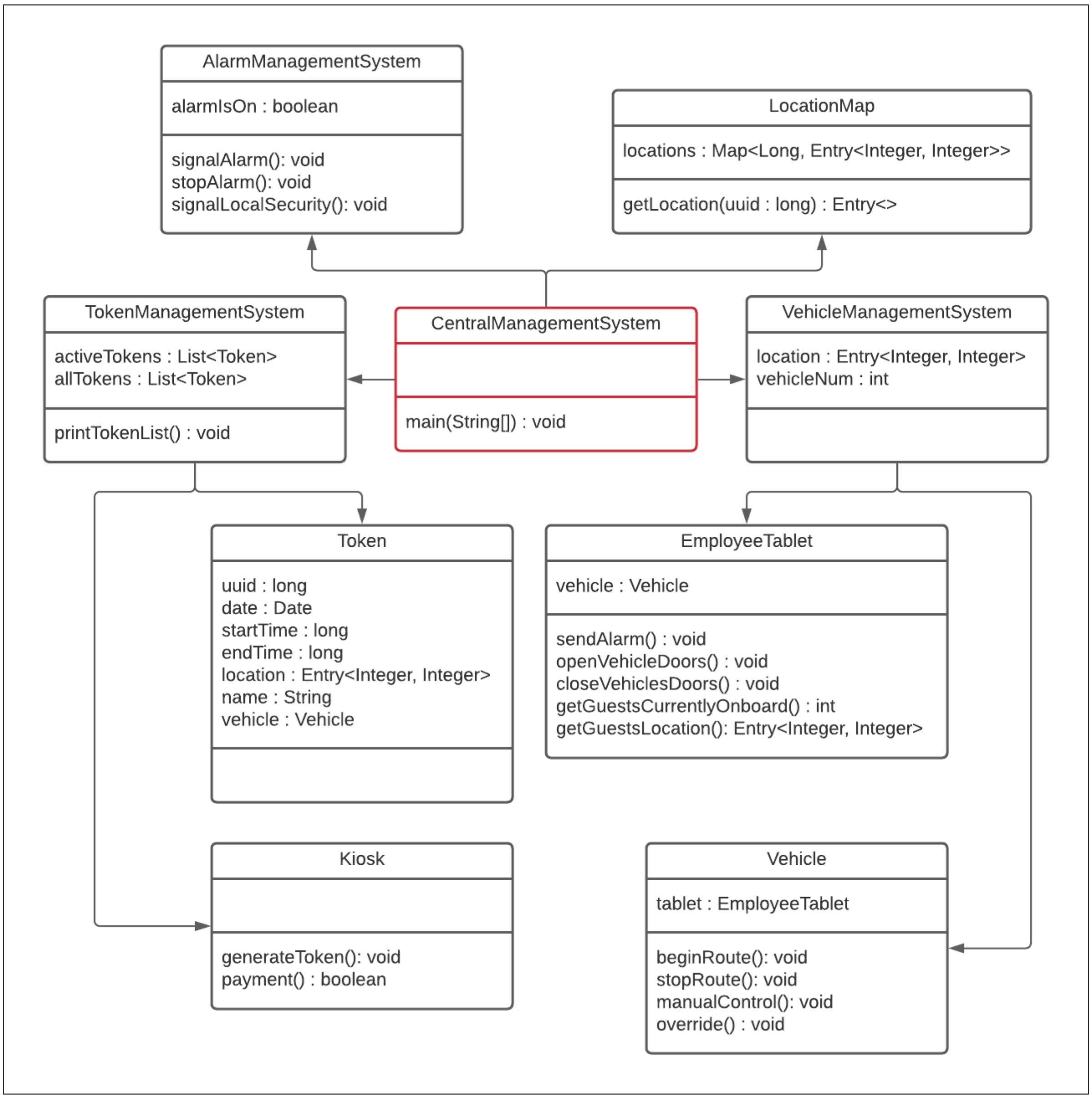

Figure 1: Class Diagram

3 Component Specifications

The following diagram is a general outline of the system which divides its functionality into its core components, which themselves are responsible for some important subsystems. The reader will notice that the components shown in this diagram do not share a one-to-one correspondence with the nomenclature of the already discussed subsystems. For example, in the diagram is the "Security System", which is not the same as the Alarm Management System. Instead, this component comprises important security features found throughout the park, of which the Alarm Management and other systems may make use. Similarly, there is no single Token Management System component shown in this diagram; rather, the components that this system controls are separated into "Token" and "Kiosk" components.

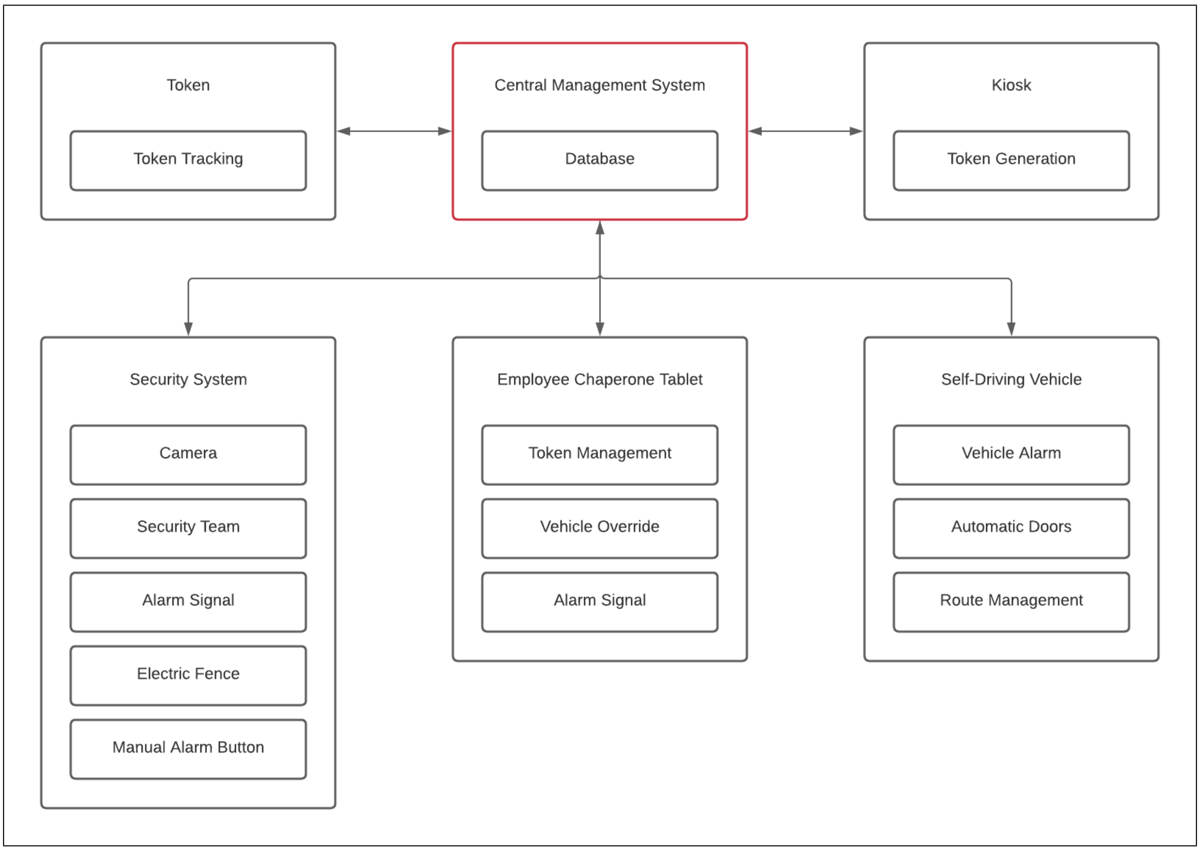

Figure 2: Component Specification Diagram

4 Sample Use Case

This section contains a collection of several convenient use-case diagrams, each of which depicts the actions that each of the different types of users can take when interacting with the system. The three types of users are the administrators, the employee chaperones, and the park visitors, each of which has its own use-case diagram as well as accompanying scenario.

The diagrams are formatted from left to right as follows: first, on the far left is depicted one of the above-mentioned types of system users; then, arrows pointing rightward indicate some of the various actions that this user may perform; from these actions may either follow intermediate side-effects, or the result of these actions may be directly reported to its corresponding subsystem. The subsystems are the actors depicted on the right sides of the diagrams.

Each diagram is color-coded to emphasize its association with one particular type of system user. Administrator actions are colored in red; employee chaperone actions in blue; and park visitor actions in yellow. Neither the diagrams themselves nor the scenarios accompanying them are to be taken as exhaustive explanations of system functionality.

4.1 Administrator Use-Case Diagram

The below diagram highlights several of the actions available to an administrator at any given time. The following depicts a scenario in which an emergency situation has arisen, and it has become necessary to force each vehicle to return to base.

The alarm system has sent information back to the central management system that some emergency has occurred. The administrator on duty at the central management system will note this, evaluate the severity of the emergency, and devise an appropriate response. In this case, they have deemed that all vehicles currently operating within the park should immediately return to base. They then press the "vehicle override" button, thereby activating the override button located on the vehicles themselves. After the administrator communicates to the employee chaperones that they should return to base, the chaperones may then press this on-board button to resume their route back to base, if they were already en route.

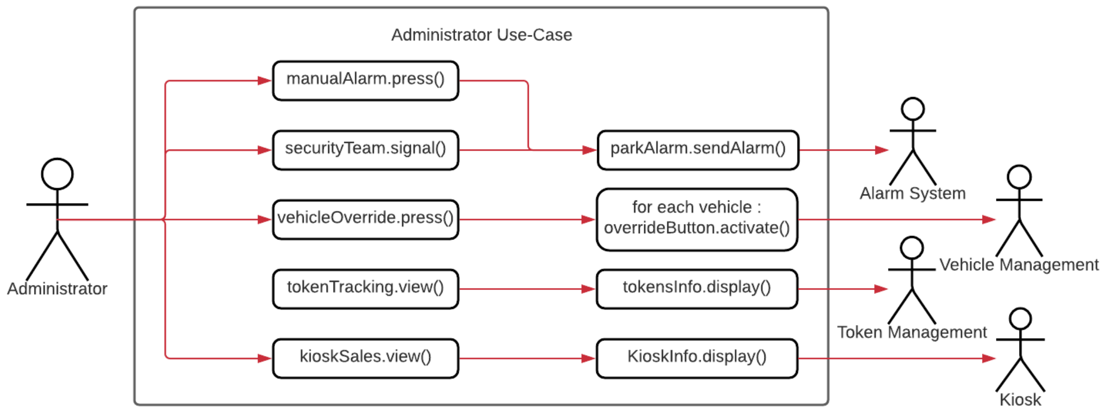

Figure 3: Use-Case Diagram for System Administrators

4.2 Employee Chaperone Use-Case Diagram

Following the scenario from the previous subsection, the below diagram depicts the course of action that an employee chaperone should take in this kind of an emergency. After receiving word from their superiors that all vehicles are ordered to return to base, the employee will press the override button located near the door of the vehicle. Since the administrator had already activated it remotely, the override button will function to close the vehicles doors and resume driving back to base.

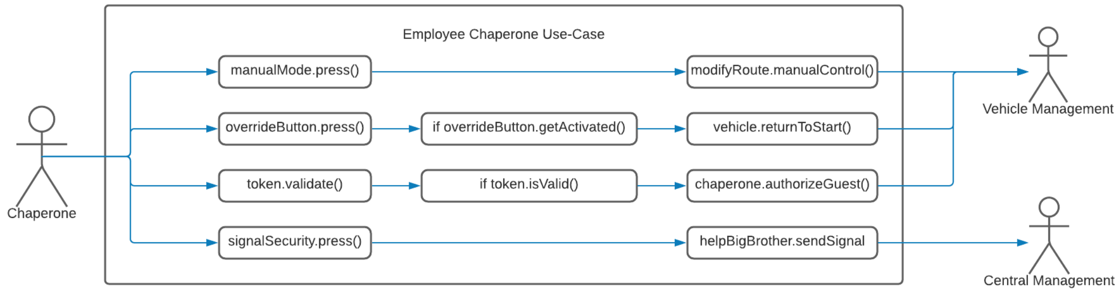

Figure 4: Use-Case Diagram for Employee Chaperones

4.3 Visitor Use-Case Diagram

The diagram below shows the interaction between a park visitor and the kiosk system located at the entrance to the park. In this scenario, a guest wishes to purchase some number of tokens so that they and their party may enjoy the park. They walk up to an available kiosk, tap the "purchase tokens" button, and are then prompted for the number of tokens they wish to purchase. The indicate such, and are then prompted for payment. They use a debit or credit card, wait for the transaction to process, and, upon successful payment, tokens are dispensed in the amount requested from the kiosk.

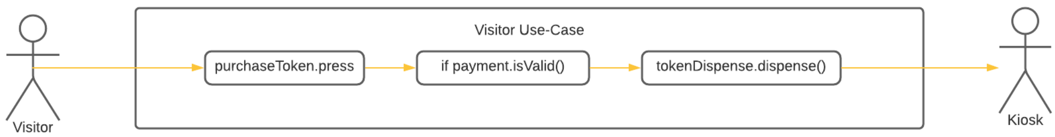

Figure 5: Use-Case Diagram for Park Visitors

5 Design Constraints

This section of the document details certain software, hardware, and security constraints which limit the functionality of the Siesta Gardens Controller. These constraints are those which affect any of the subsystems or components discussed above.

5.1 Software Constraints

Software constraints are those which limit either the functionality or performance of the software architecture that controls the SGC.

Subsystem Interdependence. All of the subsystems of the SGC are connected in some way to the Central Management System. As such, they all exhibit some degree of interdependence with each other, either directly or indirectly. Because of this, if any individual subsystem fails, then the functionality of the whole system is compromised.

Architecture Language. The implementation of the SGC will be done using Java and its associated graphics library JavaFX. The latest available version of the ZuluJDK (the normal JDK bundled together with the JavaFX libraries) will used; therefore, the SGC will not support systems without these latest versions.

Self-Driving Vehicles. The self-driving vehicles will be purchased from third parties. Therefore, the implementation of their driving software is the responsibility of the vehicle manufacturer, and not the SGC. Compatibility with the SGC is of utmost importance when selecting the vehicles to use within the park.

5.2 Hardware Constraints

The SGC is monitored from a central hub located near the entrance to the park. From this hub, an administrator will be provided with a screen that displays information about all of the SGC's subsystems. The administrator interacts with the SGC through mouse and keyboard.

Manual Alarm. The manual alarm can only be activated through physical interaction, so an administrator must be present. In a situation where the manual alarm cannot be activated, the result could be catastrophic. Without regular maintenance to the device, there is a chance that the device may not function properly. Park staff should be aware of the importance of this device and are responsible for its maintenance.

Network. Subsystems of the SGC are wirelessly connected so the quality of the network they are connected to is a major priority. In a situation where the network is damaged or interrupted, the overall system may fail to operate properly and this could put visitors in danger. Before opening the park, the network needs to be tested to ensure that all the subsystems will function properly. The park should remain closed until the administrator is confident in the network's stability.

RFID. All of the tokens operate through the use of an RFID chip and as such the tokens have a range of one thousand meters. If the token is too far away from the sensors then it cannot be read, and the smooth operation of the vehicle management system is thus not guaranteed.

Database. A database failure can be effected in many ways, including memory errors, system crashes, and misuse. If a database failure occurs, it will severely limit the functionality of the SGC as it is responsible for storing important information about park visitors.

Kiosk. The park kiosks must be built such that they are resistant to negative weather conditions so as to prevent malfunction year-round. The kiosks will only accept credit or debit card payments from park visitors. The kiosks should be monitored by an employee throughout the day to ensure that the machines are operating properly. If not monitored and a kiosk fails, then the risk of losing customers may arise. In general, additional kiosks should be available to replace malfunctioning Kiosks.

Vehicles. Self-driving vehicles require the park to have well maintained roads to prevent damage to the vehicle and for the safety of the visitors. Failure to maintain the roads may result in the failure of the vehicle, which limits the smooth operation of the park. The capacity per vehicle is limited to a maximum of ten visitors. Each vehicle should have the override button conveniently placed where a guest or employee can access it in an emergency. Trained mechanics should be available to perform regular maintenance on the vehicles. Failure to perform regular maintenance on the vehicles will hinder the operation of the park, and may even jeopardize guest safety.

Tablets. Employee Chaperone Tablets are components of the Vehicle Management System, and, as such, device failure will limit the functionality of the park's other subsystems. The devices run on a limited battery so unless regularly recharged and maintained, they will be unable to function properly. In addition, the software used on the tablets requires some technical competence on the part of its user.

5.3 Security Constraints

The SGC is responsible for activating emergency alarms around the park and safely escorting visitors back to the barge that transported them to the island. In situations where the T-Rex is able to escape its enclosure, the SGC is not responsible for re-containment.

The administrator of the SGC must have the ability to signal for local park security for small scale issues; however, the SGC is not responsible for the management of the local park security team.

Additionally, many different types of users will interact with the SGC, though mainly the administrator and employees with chaperone tablets. As a result, improper use of SGC subsystems may result in false alarms and other accidents if not properly monitored. Staff are expected to be well disciplined to prevent unauthorized access to these subsystems.

The override button's ease-of-access may make it prone to improper or unintended usage. To combat this, the Central Management System or the Alarm Management System must first raise a park-wide emergency state, whereafter the override buttons become usable. Prior to the activation of such a state, these buttons will not function.