**Document 03 – Sprint 1 Report**

CS 4321 – Summer 2022

Contents

[1 Sprint 1 Backlog 1](#_Toc106028683)

[2 Class Diagram 2](#_Toc106028684)

[3 System Tests 2](#_Toc106028685)

[4 Sprint 1 Presentation 2](#_Toc106028686)

# Sprint 1 Backlog

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Deliverable**  Develop a set of user stories that you can accomplish in the form of a working system by the end of the sprint (July 11). Note that “Status” is one of these values:   |  |  | | --- | --- | | **Status** | **Description** | | Complete | System tests verify that story has been implemented correctly. | | Buggy | Sometimes works according to system tests | | Incomplete | Started but not completed | | Not Implemented | No code written. | |

|  |  |
| --- | --- |
| US Number | 1 – Complete |
| User Story (US) | As a key office administrator, I want to be able to save building information to a text file. |
| Requirement | 1 |
| Notes |  |

|  |  |
| --- | --- |
| US Number | 2 – Complete |
| User Story | As a key office administrator, I want to be able to save employee information to a text file. |
| Requirement | 1 |
| Notes |  |

|  |  |
| --- | --- |
| US Number | 3 – Incomplete |
| User Story | As a key office administrator, I want to be able to save access information to a text file |
| Requirement | 1 |
| Notes |  |

|  |  |
| --- | --- |
| US Number | 4 – Complete |
| User Story | As a key office administrator, I want to be able to add buildings. |
| Requirement | 2 |
| Notes | Building codes should be 2 digits long. |

|  |  |
| --- | --- |
| US Number | 5 – Complete |
| User Story | As a key office administrator, I want to be able to add suites. |
| Requirement | 2 |
| Notes | Suite codes should be 2 digits long. |

|  |  |
| --- | --- |
| US Number | 6 – Complete |
| User Story | As a key office administrator, I want to be able to add rooms. |
| Requirement | 2 |
| Notes | Room numbers should be 3 digits long. |

|  |  |
| --- | --- |
| US Number | 7 – Complete |
| User Story | As a key office administrator, I want to be able to add employees. |
| Requirement | 3 |
| Notes | Employee IDs must be 4 digits long. |

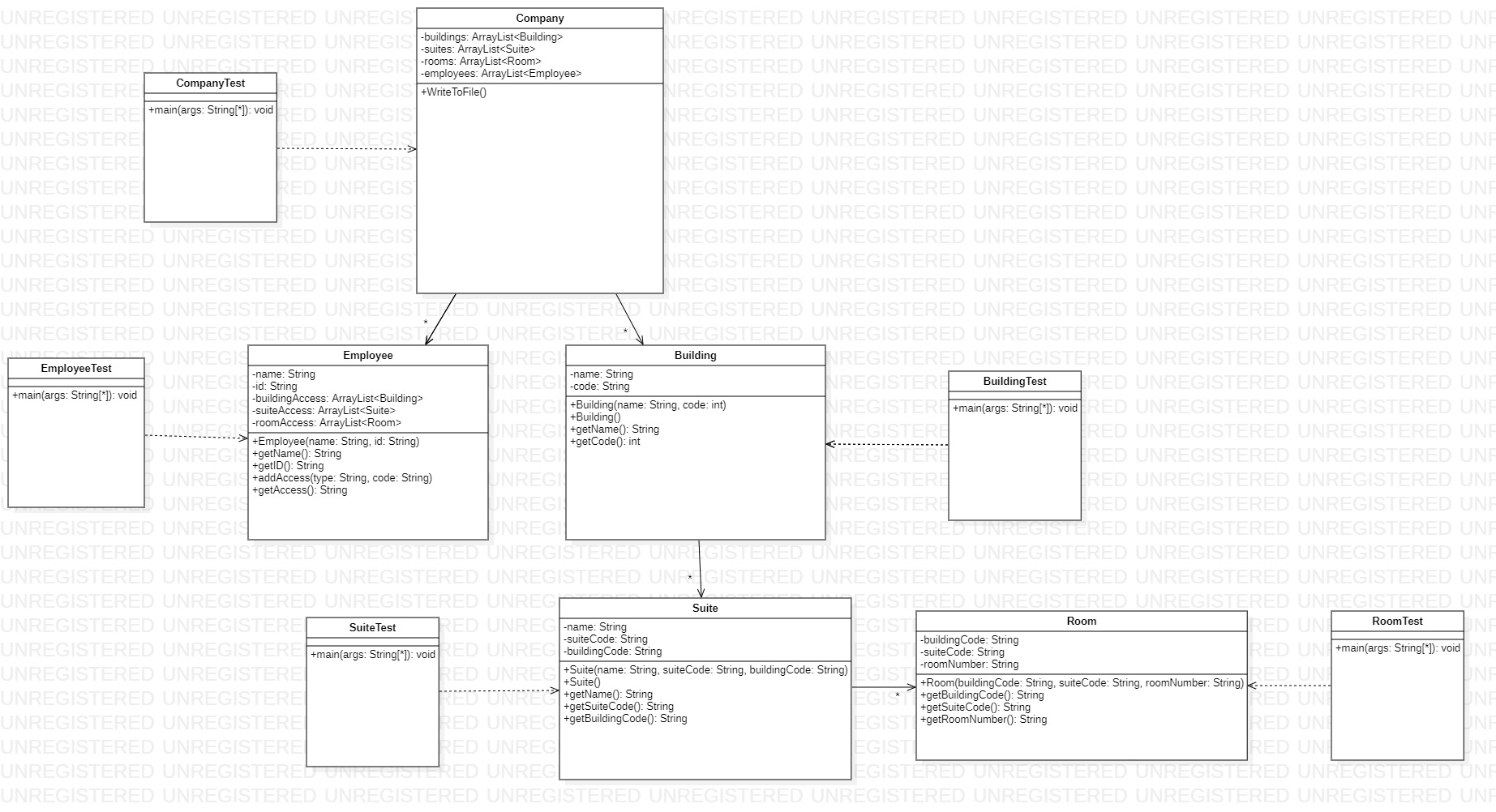
|  |  |
| --- | --- |
| US Number | 8 – Complete |
| User Story | As a key office administrator, I want to be able to add employee access to buildings, suites, and rooms. |
| Requirement | 4 |
| Notes |  |

# Class Diagram

|  |
| --- |
| **Deliverable**  At the conclusion of this sprint, you should make a class diagram using StarUML. You can make multiple diagrams at different levels of granularity, or just break it up. You objective is to convey your design to me, thus some brief remarks would be useful. The diagram(s) must be readable. Also include the diagrams saved as image files in your docs folder on GitHub. |

As we can see in our UML diagram, the Company class has many Employees and Buildings. The Building class has many Suites and the Suite class has many Rooms.

Employee is a class that represents company employees. The class is responsible for storing employee names, IDs, and areas they have access to. The Room class represents the rooms that can be found inside the company's buildings. The Room class stores room numbers and tracks what suite and building a room belongs to. The Suite class represents the suites of a building. Each suite stores its name, suite code, and the building it is inside. The Building class represents buildings which everything is within. It stores a building’s name and building code.



# System Tests

|  |
| --- |
| **Deliverable**  You probably will have multiple System Tests for each user story. |

|  |  |
| --- | --- |
| Test Number | 1 |
| US Number | 1 |
| Description | Using a class that can create txt file this test checks if the class properly saves building information to itself before generating a txt file |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 1 |
| US Number | 2 |
| Description | Using a class that can create txt file this test checks if the class properly saves employee information to itself before generating a txt file |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 1 |
| US Number | 6 |
| Description | Testing if an error is thrown when room number is missing from the constructor instead of supplying a building, suite, and room number. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 2 |
| US Number | 6 |
| Description | Testing if an error is thrown when suite is missing from the constructor instead of supplying a building, suite, and room number. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 3 |
| US Number | 6 |
| Description | Testing if an error is thrown when building is missing from the constructor instead of supplying a building, suite, and room number. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 4 |
| US Number | 6 |
| Description | Testing if an error is thrown when building code has less digits than it should. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 5 |
| US Number | 6 |
| Description | Testing if an error is thrown when suite code has less digits than it should. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 6 |
| US Number | 6 |
| Description | Testing if an error is thrown when room number has less digits than it should. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 7 |
| US Number | 6 |
| Description | Testing if an error is thrown when room number has more digits than it should |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 8 |
| US Number | 6 |
| Description | Testing if an error is thrown when room number has more digits than it should |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 9 |
| US Number | 6 |
| Description | Testing if an error is thrown when room number has more digits than it should |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 10 |
| US Number | 6 |
| Description | Testing if constructor properly saves instance variables. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 1 |
| US Number | 5 |
| Description | Testing if an error is thrown because constructor has empty strings. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 2 |
| US Number | 5 |
| Description | Testing if constructor saves instance variables correctly. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 3 |
| US Number | 5 |
| Description | Testing if getName() method correctly retrieves the name variable. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 4 |
| US Number | 5 |
| Description | Testing if getSuiteCode() method correctly retrieves the suiteCode variable. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 5 |
| US Number | 5 |
| Description | Testing if an error is thrown because format of the suiteCode variable is too long. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 6 |
| US Number | 5 |
| Description | Testing if an error is thrown because format of the suiteCode variable is too short. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 7 |
| US Number | 5 |
| Description | Testing if an error is thrown because format of suiteCode variable has invalid characters. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 8 |
| US Number | 5 |
| Description | Testing if getBuildingCode() method correctly retrieves the buildingCode variable. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 9 |
| US Number | 5 |
| Description | Testing if an error is thrown because the format of the buildingCode variable is too long. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 10 |
| US Number | 5 |
| Description | Testing if an error is thrown because format of the buildingCode variable is too short. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 11 |
| US Number | 5 |
| Description | Testing if an error is thrown because format of the buildingCode variable has invalid characters. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 1 |
| US Number | 7 |
| Description | Testing if an error is thrown because of an empty string was submitted for employee name |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 2 |
| US Number | 7 |
| Description | Testing if an error is thrown because of an empty string was submitted for employee ID. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 3 |
| US Number | 7 |
| Description | Testing if an error is thrown because the ID supplied in constructor has more than 4 digits |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 4 |
| US Number | 7 |
| Description | Testing if an error is thrown because the ID supplied in constructor has less than 4 digits |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 5 |
| US Number | 7 |
| Description | Using getters to test if constructor properly saves instance variables |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 1 |
| US Number | 8 |
| Description | Testing if an error is thrown because building code submitted for access is less than 2 |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 2 |
| US Number | 8 |
| Description | Testing if an error is thrown because building code submitted for access is greater than 2 |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 3 |
| US Number | 8 |
| Description | Testing if an error is thrown because suite code submitted for access is less than 2 |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 4 |
| US Number | 8 |
| Description | Testing if an error is thrown because suite code submitted for access is greater than 2 |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 5 |
| US Number | 8 |
| Description | Testing if an error is thrown because room number submitted for access is less than 3 |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 6 |
| US Number | 8 |
| Description | Testing if an error is thrown because room number submitted for access is greater than 3 |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 7 |
| US Number | 8 |
| Description | Testing if access can be given to just buildings. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 8 |
| US Number | 8 |
| Description | Testing if access can be given to just suites. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 9 |
| US Number | 8 |
| Description | Testing if access can be given to just rooms. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 10 |
| US Number | 8 |
| Description | Testing if access can be given to just buildings and suites |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 11 |
| US Number | 8 |
| Description | Testing if access can be given to just buildings and rooms |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 12 |
| US Number | 8 |
| Description | Testing if access can be given to just suites and rooms. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 13 |
| US Number | 8 |
| Description | Testing if access can be given to buildings, suites, and rooms at the same time. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 1 |
| US Number | 4 |
| Description | Testing if the getName function works. |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 2 |
| US Number | 4 |
| Description | Testing if getCase for the building class works under expected conditions |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 3 |
| US Number | 4 |
| Description | Testing if getCase for the building class works when there are more than 2 digits |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 4 |
| US Number | 4 |
| Description | Testing if getCase for the building class works when there are less than 2 digits |
| Status | Success |

|  |  |
| --- | --- |
| Test Number | 5 |
| US Number | 4 |
| Description | Testing if getCase for the building class works when there are letters entered instead of strings |
| Status | Success |

# Sprint 1 Presentation

* It is preferred that all team members be present for the presentation.
* Your completed Sprint 1 Report should be in your GitHub docs folder by 7am on July 11 so that I can look at it before the presentation.
* Before your presentation begins, have these things open: your code in an IDE, your Sprint 1 Report, your class diagram.
* The presentation will be interactive, meaning I’ll stop you and ask questions.
* You’ll begin by doing these things:

1. Explain your design. Show your class diagram(s) and explain. Do: (a) explain what the major purpose is of each class (“this class represents…”, (b) how classes are related (has-a, or has-many, or is-a), (c) any major decision choices you made, if applicable. Don’t: (a) mention every method and instance variable – you can mention a method or two, if that helps explain things.
2. Show me 3 interesting unit tests and briefly explain.
3. Demo your system for about 10 minutes. State the user story and then demo. Repeat. You don’t have to do them all, but be prepared to if asked.

Note: Your objective in a demo is to clearly show that your system works as intended. Thus, do NOT move to quickly, make sure the audience (me) is following, remind if things need to be remembered to understand a current step, etc. You’ve likely run your software a zillion times and move very, very quickly through the software when you are running it to test. Don’t do that in a demo!