Software Engineering Group Project

By: Michael Giraldi, Christopher Pando, Edward Aranda, Jordan Adriano

Dr. Shengli Yuan

CS3321-22735

November 1, 2020

**Table of Contents**

[**Requirements**](#_kwsvyqearq3r) **7**

[**Modification History**](#_86zwtld78363) **8**

[**Domain Knowledge**](#_sn3eyopqxfaf) **8**

[Glossary](#_8dymgj5vt09v) 9

[Interview with Client](#_t1e1t3ues2z2) 10

[**Functional Requirements**](#_wtxhtk4kt6o1) **12**

[Use Cases](#_8jhodrfqrsj0) 12

[Use Case 1](#_29sb5mxa7v88) 12

[Use Case 2](#_a03jusxrraj4) 12

[Use Case 3](#_a5fiprlg4679) 13

[Use Case 4](#_ea05kohia5at) 13

[Use Case 5](#_nc1ovxxcqq1z) 14

[Use Case 6](#_e6d2urbe4mg5) 15

[Use Case 7](#_1qf73pdu1a0s) 15

[Use Case 8](#_hve3c2l9rtyb) 16

[Use Case 9](#_7sttbgu1d56h) 16

[Use Case 10](#_jmmhuobq4fw5) 17

[Use Case 11](#_wdv9odj35fy9) 17

[Use Case 12](#_450ttcb96feq) 18

[Use Case 13](#_9fwupxogodu5) 18

[Use Case 14](#_5g1vcygc5jf) 19

[Use Case 15](#_g8dcld77m2od) 20

[Use Case 16](#_olenj2tt2g6y) 20

[Use Case 17](#_kbgsjcmxckdu) 21

[Use Case 18](#_bgazged3oq1v) 21

[Use Case 19](#_rr4pz04la9sw) 22

[Use Case 20](#_xfny66l0j6do) 23

[Use Case 21](#_yqyhcby4qg7) 23

[**Non-Functional Requirements**](#_l7d6yqas03rd) **25**

[Cost Constraints](#_u29uft7fehml) 25

[Reliability](#_hdy6rpduptsy) 25

[Time Constraints](#_y3403hpwsuc3) 25

[**Product Specifications**](#_on2h5y58g2hc) **26**

[**Modification History**](#_rjd8oxv8t50u) **27**

[**Major Milestones**](#_ukd8dwapbv5z) **28**

[**Abstract**](#_67h1t89vbrqd) **29**

[**Document References**](#_rsdglny0wpoq) **30**

[**Glossary**](#_7fh0pt1medps) **31**

[**Class Diagram**](#_ao2kwzquvnh4) **32**

[**Use Case Sequence Diagram**](#_982e8o4fgpi2) **33**

[Use Case 1](#_axp1841qrvvb) 33

[Use Case 2](#_iq5ge3km894b) 33

[Use Case 3](#_mtvuq1b088r0) 34

[Use Case 4](#_1sgdugt42aw7) 34

[Use Case 5](#_h4b80p7bh6zv) 35

[Use Case 6](#_n6m0pjdwhqdg) 36

[Use Case 7](#_l52yzo74utq4) 36

[Use Case 8](#_q1h4hb6ecjo3) 37

[Use Case 9](#_dmee4o4cuvs8) 38

[Use Case 10](#_3j2il7gs61jf) 38

[Use Case 11](#_g0agh9u1i6zs) 39

[Use Case 12](#_vxai0istsibq) 40

[Use Case 13](#_omz5gnjlrxe0) 41

[Use Case 14](#_pk7z3vmlx2g4) 42

[Use Case 15](#_gh32xfrro6y1) 42

[Use Case 16](#_g0g2eqhl8c1e) 43

[Use Case 17](#_cctjmzwe4um3) 43

[Use Case 18](#_dlr031ertfi) 44

[Use Case 19](#_sby7gzf6ry54) 45

[Use Case 20](#_j9wrf85jlofp) 45

[Use Case 21](#_7517q9rpvlyg) 46

[**User Interface**](#_fivkq6len8k6) **50**

[Picture/Description](#_11tp0p4gs18c) 50

[**Others**](#_qy09tfanjxfy) **55**

[**Software Project Management Plan**](#_v3j4g1t37h8) **56**

[**PREFACE**](#_my2xps409bj4) **57**

[**1. Overview of the Product**](#_dsponikwzezb) **57**

[Purpose Scope & Objective](#_ccf72h430s7s) 57

[1.1 Project Background](#_ccf72h430s7s) 57

[1.2 Assumptions & Constraints](#_r2jognabrgq) 57

[1.3 Project Deliverables](#_ccf72h430s7s) 57

[1.4 Schedule & Budget Summary](#_wtobnr6ppckj) 58

[1.5 Evolution of the Plan](#_wtobnr6ppckj) 58

[**2. References**](#_i7rv2owsz4yb) **59**

[**3. Definitions & Acronyms**](#_82n9azkfk53s) **59**

[**4. Project Organization**](#_6aiioao5z4jh) **61**

[4.1 External Interfaces](#_ejovmaev9coj) 61

[4.2 Internal Structure](#_ejovmaev9coj) 61

[4.3 Roles & Responsibilities](#_ejovmaev9coj) 61

[4.4 Team Methods of Operation](#_n600tnv9471) 62

[**5. Managerial Process Plans**](#_z4eymrx4m46b) **62**

[5.1 Start-Up Plan](#_d7jirms6wtzg) 62

[Training](#_txxuxoo6mag2) 63

[5.2 Work Plan](#_xn3eyqbus593) 63

[5.2.1 Work Activities](#_8zho4ptnbn3) 63

[5.2.2 Schedule Allocation](#_etu7zp5y0yjj) 65

[5.2.3 Resource Allocation](#_etu7zp5y0yjj) 65

[5.2.4 Budget Allocation](#_xhd66shi2n0v) 66

[5.3 Control Plan](#_m8qhdaibq5s7) 66

[5.3.1 Requirements Management](#_hpnkr9gmfcqt) 66

[5.3.2 Schedule Control](#_3fx1a799gzik) 68

[5.3.3 Resource Control](#_3fx1a799gzik) 68

[5.3.4 Budget Control](#_3fx1a799gzik) 68

[5.3.5 Reporting & Communication Plan](#_3fx1a799gzik) 68

[5.3.6 Measurement Plan](#_i2qjiujwmkk0) 69

[5.4 Risk Management Plan](#_sp0loqle0rl4) 70

[Risk Management Note](#_h2h70c91v239) 71

[5.5 Closeout Plan](#_uokqvg73emoz) 71

[**6. Technical Process Plans**](#_a3j5m0bdv9w5) **71**

[6.1 Process Model](#_map35giv0znt) 71

[6.2 Methods, Tools, & Techniques](#_nhgj5s5jwbne) 72

[Programming Environment](#_mmrzg8lsykxb) 72

[Database Environment](#_mmrzg8lsykxb) 72

[Version Control](#_mmrzg8lsykxb) 72

[Documentation](#_mmrzg8lsykxb) 73

[Testing](#_mmrzg8lsykxb) 73

[6.3 Infrastructure Plan](#_wrgh1ztfo4n7) 73

[6.4 Product Acceptance Plan](#_qyj2l5cj7eox) 73

[6.5 Deployment Plan](#_wrgh1ztfo4n7) 73

[**7. Supporting Process Plans**](#_tan8mdsofos6) **73**

[7.1 Configuration Management Plan](#_2gv78xi4crbo) 73

[7.2 Product Testing & Reviews Plan](#_2gv78xi4crbo) 73

[7.3 Document & Work Product Plan](#_2gv78xi4crbo) 74

[7.4 Quality Assurance Plan](#_bvnucggkorr) 74

[7.5 Project Progress Reviews](#_bvnucggkorr) 75

[7.6 Issue Management](#_bvnucggkorr) 75

[7.7 Version Management](#_m4nvbb13upht) 76

[7.8 Subcontract Management (Acquisition Management) Plan](#_rxdcrm2n7zrh) 77

[7.9 Process Improvement Plan](#_ev48jlb8k363) 77

[**Document Control**](#_z3cx8oekv17b) **77**

[Change History](#_eh14dvvo3wfq) 77

[Document Storage](#_g6v8mudze8bn) 79

[Document Owner](#_q9u4x0ydf0vk) 79

[**Appendices**](#_ltas66yix6ff) **79**

[**Detailed Design**](#_4xns48tpy017) **80**

[**Functions (Pseudocode)**](#_yvec1sy02cll) **81**

[Activity Log](#_68me80ca1ipz) 81

[Departments](#_nqo4bf6o45s5) 81

[Employee](#_qwstkxj1kar2) 81

[Login](#_kn2j49gao3m8) 82

[Main](#_rgzh7psgeee) 82

[**Software Quality Assurance Plan (Test Plan)**](#_i8liqlionz63) **83**

[**Document History and Distribution**](#_sc6aqtb41z8e) **84**

[1. Revision History](#_30m3p627fa04) 84

[2. Distribution](#_7wwzdjrggh1d) 85

[**1. Introduction**](#_g2z8vgln6n7e) **86**

[1.1 Objectives](#_qy431b26yu8h) 86

[1.2 Testing Strategy](#_587r6vbs9fh) 86

[1.3 Scope](#_7w1t92kydnjg) 86

[1.4 Reference Material](#_4kszy91th2nk) 86

[1.5 Definitions and Acronyms](#_nei1ajijpbtf) 87

[Glossary](#_v07z4q6am3b0) 87

[**2. Test Items**](#_iheticriar58) **87**

[2.1 Program Modules](#_m8qki6n5m65e) 88

[2.2 Job Control Procedures](#_4vfq977mwk4y) 88

[2.3 User Procedures](#_paiyi4hzwun4) 88

[2.4 Operator Procedures](#_yev39dyfh6ql) 88

[**3. Features To Be Tested**](#_fqrp30iepxqi) **89**

[**4. Features Not To Be Tested**](#_cgjhuq2w9g4y) **90**

[**5. Approach**](#_hamocfpubze1) **91**

[5.1 Component Testing](#_w33u5lmcrm00) 91

[5.2 Integration Testing](#_qg57xyu9cmvb) 91

[5.3 Conversion Testing](#_a12dsjj0b5is) 91

[5.4 Job Stream Testing](#_3sfa7c22sxe8) 92

[5.5 Interface Testing](#_cvvcnb6mjxe2) 92

[5.6 Security Testing](#_dn8vbzbw7pf2) 92

[5.7 Recovery Testing](#_ha6nd1xnmfx4) 92

[5.8 Performance Testing](#_ao6rt8lg2w7f) 92

[5.9 Regression Testing](#_ojr9wo5skcaz) 92

[5.10 Acceptance Testing](#_nqwkak7ztjk) 93

[5.11 Beta Testing](#_v61kg42d8l0s) 93

[**6. Pass / Fail Criteria**](#_otn08qpdnpbv) **93**

[6.1 Suspension Criteria](#_yg8cf9wwqrs1) 93

[6.2 Resumption Criteria](#_3edgn7uepkj) 93

[6.3 Approval Criteria](#_55vs1huodtg5) 94

[**7. Testing Process**](#_xhm6wrxx6g38) **94**

[7.1 Test Deliverables](#_780wn6tuttam) 94

[7.2 Testing Tasks](#_6ts7do1aihuf) 95

[7.3 Responsibilities](#_dodsr8pxp1br) 95

[7.4 Resources](#_6y4vhgpxlnx7) 95

[7.5 Schedule](#_qi29ixxyrm8h) 96

[**8. Environmental Requirements**](#_v7yl4avjr4xo) **96**

[8.1 Hardware](#_dg5uat8v4efv) 96

[8.2 Software](#_g5dasbnqo19e) 96

[8.3 Security](#_dn9e7n3rt9br) 96

[8.4 Tools](#_6cxknupyb3oh) 96

[8.5 Publications](#_ek1tk1rn1xrx) 97

[8.6 Risks and Assumptions](#_7hx2jblaefqy) 97

[**9. Change Management Procedures**](#_wgpsvae2m5gi) **97**

[**10. Plan Approvals**](#_g40eph2u64ay) **97**

# Requirements

## Modification History

|  |  |  |
| --- | --- | --- |
| **Date** | **Modified By** | **Changes** |
| 10/14/2020 | Edward, Christopher, Michael, Jordan | * Added Functional Requirements |
| 10/16/2020 | Edward, Christopher, Michael | * Added Interview, Non-Functional Requirements |
| 10/20/2020 | Christopher, Michael | * Modified Use Cases |
| 10/23/2020 | Edward, Christopher, Michael, Jordan | * Added Glossary * Modified Use Cases |
| 10/27/2020 | Edward, Christopher, Michael | * Modified Glossary, Use Cases, Non-Functional Requirements |
| 10/30/2020 | Christopher, Michael | * Modified Use Cases |
| 10/31/2020 | Michael | * Modified Use Cases |

## Domain Knowledge

### Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Void | To cancel a clients order. |
| Edit | To change a clients order. |
| Markdown | A reduction in price. |
| Stock | The goods or merchandise kept on the premises of a business or warehouse and available for sale or distribution. |
| Order | The summary of items a client ordered. |
| Log | An official record of events |

|  |  |
| --- | --- |
| Acronym | Meaning |
| POS (Point of Sale) | The point of sale or point of purchase is the time and place where a retail transaction is completed. |
| CRO (Conversion Rate Optimization) | Conversion rate optimization is a set of tactics ecommerce companies use to improve the ratio between site visits and conversions (downloading their apps, signing up for newsletters). |
| SH | Stan’s Hardware |
| DOB | Date of Birth |

### Interview with Client

**Location**: Stan's Hardware

**Date**: October 14, 2020

**Time**: 9:00 am

**Attendees**: 3

**Description**:

Question 1: “Tell me about your business”

Response: We run a medium-large hardware store. We have 30 employees serving between 600 and 1,200 customers daily. We process over 400 transactions daily. We operate from 6 am until 10 pm.

Question 2: “How many employees access your inventory?”

Response: 23 employees including 5 managers need access to update our inventory.

Question 3: “How often do your employees access your inventory software?”

Response: At any given moment we have 3 to 4 employees accessing our inventory. In the mornings when we receive new stock we have between 2 and 4 employees frequently updating inventory.

Question 4: “How often does your inventory interact with POS devices?”

Response: We have between 300 transactions on a slow day and 1,000 transactions on a very busy day.

Question 5: “Why is your current software unsatisfactory?”

Response: Our employees are using a Google spreadsheet to update inventory. We have had instances of employees erroneously entering wrong data into the spreadsheet. This has resulted in lost time because we need to suspend employee access to inventory while the error is corrected.

Question 6: “What aspects of your current software work well?”

Response: Google sheets allows multiple people to access the same sheet. It also doesn’t let other people access the sheet unless we give them permission. When the software updates it updates it for everyone immediately.

Question 7: “How will your business be affected if this project does not move ahead?”

Response: We are losing money because our employees have too many privileges to access our inventory. We have had to close our business four days this year on separate occasions due to losing data from an employee deleting the sheet and in another case because a manager had a grudge. We had to let this manager go, and because he created the sheet we lost access to our inventory until our lawyers convinced him to restore access.

Question 8: “What are your expectations?”

Response: We want a database that our employees can’t possibly screw up. We don’t want to have a disgruntled manager be able to muff up our database. We don’t need employees stealing from us either by changing the inventory, but they need to be able to see what’s in stock.

Question 9: “What is your budget and when do you want to start?”

Response: We have a budget of $67,000 and we want you to start as soon as possible.

## Functional Requirements

### Use Cases

#### Use Case 1

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Add Supervisor |
| Tasks | Enter into Supervisor Menu, Select “Add Supervisor,” Fill in all fields, Select “Submit” |
| Preconditions | Logged in as Admin |
| Exceptions | Invalid characters used in name |
| Variation of action interactions | When the actor attempts to submit a Supervisor using invalid characters an error window will appear and return the actor to the form. A new Supervisor will not be added to the database. |
| System change/production | New supervisor is added to the Supervisor database |

**Step-by-Step Descriptions**:

1. User login

2. Log Activity

3. Add supervisor

#### Use Case 2

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Edit Supervisor |
| Tasks | Enter into Supervisor Menu, Select “Edit Supervisor,” Choose Supervisor from drop-down menu, Modify data, Select “Submit” |
| Preconditions | Logged in as Admin |
| Exceptions | Supervisor database empty |
| Variation of action interactions | When the actor tries to select a Supervisor from the Supervisor drop-down menu the menu will appear empty. The actor will have to add a Supervisor to the database first. |
| System change/production | Employee activity gets updated.  No change in stock by searching for the quantity of the item. |

**Step-by-Step Descriptions**:

1. User login

2. Log Activity

3. Edit supervisor

#### Use Case 3

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Delete Supervisor |
| Tasks | Enter into Supervisor Menu, Select “Delete Supervisor,” Choose Supervisor from drop-down menu, Select “Yes” from “Are you sure?” pop-up window |
| Preconditions | Logged in as Admin |
| Exceptions | Supervisor is currently logged in |
| Variation of action interactions | If Supervisor is logged in, the app must be temporarily suspended or that Supervisor must log out first to proceed deleting Supervisor. |
| System change/production | Supervisor is removed from the database. |

**Step-by-Step Descriptions**:

1. User login

2. Log Activity

3. Delete supervisor

#### Use Case 4

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Create new backup |
| Tasks | Navigate to “Backup” menu, Select “Create Backup,” Click “Yes” on pop-up window |
| Preconditions | Logged in as Admin |
| Exceptions | Connection to offsite server lost |
| Variation of action interactions | When connection is lost, contact helpdesk for offsite server to reestablish connection |
| System change/production | Entire database backed up to offsite location |

**Step-by-Step Descriptions**:

1. User login

2. Access departments

3. Hardware, Lighting, or Plumbing

4. Update Inventory

5. Create new backup

#### Use Case 5

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Restore from backup |
| Tasks | Navigate to “Backup” menu, Select “Restore from Backup,” Click “Yes” on pop-up window |
| Preconditions | Logged in as Admin |
| Exceptions | No backup exists |
| Variation of action interactions | If no backup exists, actor must create a backup |
| System change/production | Entire database will be reverted to previous backup |

**Step-by-Step Descriptions**:

1. User login

2. Access departments

3. Hardware, Lighting, or Plumbing

4. Update Inventory

5. Restore from backup

#### Use Case 6

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Clear Employee activity log |
| Tasks | Navigate to “Activity Log,” Select “Clear Log,” Click “Yes” on pop-up window |
| Preconditions | Logged in as Admin |
| Exceptions | Activity log already cleared |
| Variation of action interactions | When the activity log is already cleared and actor attempts to clear log again, an error window will appear and the activity log will not be cleared again. |
| System change/production | The database containing the Employee activity log will be wiped. |

**Step-by-Step Descriptions**:

1. User login

2. Activity log

3. Clear employee activity log

#### Use Case 7

**Actor**: Admin/Supervisor

|  |  |
| --- | --- |
| Goals of the actor | Add Employee |
| Tasks | Navigate to “Employee” menu, Select “Create Employee,” Enter information, Click “Submit” |
| Preconditions | Logged in as Admin or Supervisor |
| Exceptions | Negative DOB entered in form |
| Variation of action interactions | Ensure only positive integers are entered in DOB field |
| System change/production | A new employee is added to the Employee database |

**Step-by-Step Descriptions**:

1. User login

2. Activity log

3. Add new employee

#### Use Case 8

**Actor**: Admin/Supervisor

|  |  |
| --- | --- |
| Goals of the actor | Edit Employee |
| Tasks | Navigate to “Employee” menu, Select “Edit Employee,” Modify information, Click “Submit” |
| Preconditions | Logged in as Admin or Supervisor |
| Exceptions | Employee not in database |
| Variation of action interactions | When an Employee is not in the database, actor must create an Employee first. |
| System change/production | An Employee’s information is changed |

**Step-by-Step Descriptions**:

1. User login

2. Activity log

3. Edit employee

#### Use Case 9

**Actor**: Admin/Supervisor

|  |  |
| --- | --- |
| Goals of the actor | Delete Employee |
| Tasks | Navigate to “Employee” menu, Select “Delete Employee,” Select Employee ID from the drop-down menu, Click “Submit,” Click “Yes” on pop-up window |
| Preconditions | Logged in as Admin or Supervisor |
| Exceptions | Employee is logged in |
| Variation of action interactions | If Employee is logged in, suspend the program or log out Employee first, then attempt to delete again. |
| System change/production | An Employee is removed from the database. |

**Step-by-Step Descriptions**:

1. User login

2. Activity log

3. Delete employee

#### Use Case 10

**Actor**: Admin/Supervisor

|  |  |
| --- | --- |
| Goals of the actor | View Employee Activity Log |
| Tasks | Navigate to “Activity Log” menu, Click “Enter” |
| Preconditions | Logged in as Admin or Supervisor |
| Exceptions | Actor clicks “Clear Log” by mistake |
| Variation of action interactions | If actor clicks “Clear Log” by mistake, a window will appear. If actor is Admin, the window will display “Are you sure?” and a “Yes” and “No” button. Admin can click “No” and return to “Activity Log” menu. If actor is Supervisor, then the window will display an error message and return the Supervisor to the Activity Log as Supervisors may not clear the log. |
| System change/production | No change is made. |

**Step-by-Step Descriptions**:

1. User login

2. View employee activity log

#### Use Case 11

**Actor**: Admin/Supervisor

|  |  |
| --- | --- |
| Goals of the actor | Delete Inventory |
| Tasks | Navigate to “Inventory” menu, Select “Delete,” Select inventory to delete, Click “Enter” |
| Preconditions | Logged in as Admin or Supervisor |
| Exceptions | Inventory database empty |
| Variation of action interactions | Inventory must be added to the database first |
| System change/production | Inventory will be removed from the database |

**Step-by-Step Descriptions**:

1. User login

2. Access departments

3. Hardware, Lighting, or Plumbing

4. Delete inventory

#### Use Case 12

**Actor**: Admin/Supervisor/Employee

|  |  |
| --- | --- |
| Goals of the actor | View Inventory |
| Tasks | Navigate to “Inventory” menu, Select “View,” Click “Enter” |
| Preconditions | Logged in as Admin, Supervisor, or Employee |
| Exceptions | Actor clicks “Delete” by mistake |
| Variation of action interactions | If the actor is Employee, then an error window will appear and the actor may return to the Inventory menu and select “View” again. If the actor is either Supervisor or Admin, they may close the window containing the inventory and attempt to select “View” again. |
| System change/production | No change is made. |

**Step-by-Step Descriptions**:

1. User login

2. Access departments

3. Hardware, Lighting, or Plumbing

4. View inventory

#### Use Case 13

**Actor**: Admin/Supervisor/Employee

|  |  |
| --- | --- |
| Goals of the actor | Update Inventory |
| Tasks | Navigate to “Inventory” menu, Select “Update,” Select inventory to update, Click “Enter,” Modify inventory, Click “Submit” |
| Preconditions | Logged in as Admin, Supervisor, or Employee |
| Exceptions | Employee Actor tries to reduce item quantity. |
| Variation of action interactions | In an effort to discourage theft, Employees are not permitted to reduce the number of items available in an inventory. Only a Supervisor or Admin may manually remove items. |
| System change/production | Inventory details are modified. |

**Step-by-Step Descriptions**:

1. User login

2. Access departments

3. Hardware, Lighting, or Plumbing

4. Update inventory

#### Use Case 14

**Actor**: Supervisor/Employee

|  |  |
| --- | --- |
| Goals of the actor | Create a password |
| Tasks | From login page select “Create Account,” Enter ID and a password, Click “Create” |
| Preconditions | Must have an ID in Supervisor or Employee database |
| Exceptions | Invalid ID |
| Variation of action interactions | If actor has an invalid ID they may have entered it incorrectly and must re-enter their correct ID. Invalid ID error may also result from the ID not existing in the database. An Admin or Supervisor must manually add the Supervisor or Employee to the database first before an account can be created. |
| System change/production | A Supervisor’s or Employee’s hashed password and ID is added to the Login database. |

**Step-by-Step Descriptions**:

1. Create Account
2. Enter ID number
3. Create a password

#### Use Case 15

**Actor**: Admin/Supervisor/Employee

|  |  |
| --- | --- |
| Goals of the actor | View Transactions |
| Tasks | Navigate to “Transaction” menu, Select “View,” Click “Enter” |
| Preconditions | Logged in as Admin, Supervisor, or Employee |
| Exceptions | Actor enters the wrong menu |
| Variation of action interactions | If the actor enters the wrong menu, they must select “back” from the page they are on and re-navigate to the “Transaction” menu. |
| System change/production | No change is made. |

**Step-by-Step Descriptions**:

1. User login
2. Transaction menu
3. Select ‘View’

#### Use Case 16

**Actor**: Admin/Supervisor

|  |  |
| --- | --- |
| Goals of the actor | Delete Transaction |
| Tasks | Navigate to “Inventory” menu, Select “Delete Transaction,” Select transaction from the menu, Click “Delete” |
| Preconditions | Logged in as Admin or Supervisor |
| Exceptions | No transactions in database |
| Variation of action interactions | A transaction must exist in the transactions database to be deleted from the database. Transactions are only added to the database from POS devices and cannot be manually added to the database. |
| System change/production | A transaction is removed from the database. |

**Step-by-Step Descriptions**:

1. Navigate to Inventory menu
2. Select Delete Transaction
3. Select Transaction to delete
4. Delete transaction

#### Use Case 17

**Actor**: Supervisor/Employee

|  |  |
| --- | --- |
| Goals of the actor | Enter Customer Satisfaction Rating |
| Tasks | Navigate to “Ratings” menu, Select “Get Customer Rating,” Prompt customer to give rating, Enter rating and comments, Click “Enter” |
| Preconditions | Logged in as Supervisor or Employee  Transaction ID must exist |
| Exceptions | Transaction ID does not exist |
| Variation of action interactions | A transaction ID must exist to tie it to a customer satisfaction rating. Allow the customer to complete a purchase before attempting to enter a satisfaction rating. |
| System change/production | A customer satisfaction rating and actor’s comments are added to the Ratings database. |

**Step-by-Step Descriptions**:

1. User login

2. Employee section

3. Customer Satisfaction Survey

4. Take survey

5. Enter customer satisfaction rating

#### Use Case 18

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Delete Customer Satisfaction Rating |
| Tasks | Navigate to “Ratings” menu, Select “Delete Rating,” Select rating to delete, Click “Delete,” Click “Yes” when pop-up window appears |
| Preconditions | Logged in as Admin |
| Exceptions | Rating not found in database |
| Variation of action interactions | If a rating is not found, it may not exist. Actor must add a rating to the database before deleting it. |
| System change/production | A customer satisfaction rating is removed from the database. |

**Step-by-Step Descriptions**:

1. User login

2. Employee section

3. Customer Satisfaction Survey

4. Take survey

5. Delete customer satisfaction rating

#### Use Case 19

**Actor**: Admin/Supervisor

|  |  |
| --- | --- |
| Goals of the actor | View Customer Satisfaction Rating |
| Tasks | Navigate to “Ratings” menu, Select “View Ratings,” Click “Enter” |
| Preconditions | Logged in as Admin or Supervisor |
| Exceptions | Actor clicks on “Delete” by mistake |
| Variation of action interactions | If the actor is an Admin, they will be asked if they are sure they want to delete the rating. The actor may navigate back by closing the window. |
| System change/production | No changes are made. |

**Step-by-Step Descriptions**:

1. User login

2. Employee section

3. Customer Satisfaction Survey

4. Take survey

5. View customer satisfaction rating

#### Use Case 20

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Create Department |
| Tasks | User logs in, accesses File Dropdown menu and selects Create Department. |
| Preconditions | Logged in as Admin |
| Exceptions | Invalid character entered in department name |
| Variation of action interactions | If an invalid character is entered an error window will appear and the Admin will have to re-enter the name. |
| System change/production | A department is added to the database. |

**Step-by-Step Descriptions**:

1. Admin login
2. Access File drop-down menu
3. Select ‘Create Department’

#### Use Case 21

**Actor**: Admin

|  |  |
| --- | --- |
| Goals of the actor | Delete Department |
| Tasks | User logs in, accesses File Dropdown menu and selects Create Department. |
| Preconditions | Logged in as Admin |
| Exceptions | Department has already been deleted |
| Variation of action interactions | If the department has already been deleted no further action is required from the Admin |
| System change/production | A department is removed from the database. |

**Step-by-Step Descriptions**:

1. Admin login
2. Access File drop-down menu
3. Select ‘Delete Department’

### 

### 

### 

## Non-Functional Requirements

### Cost Constraints

Stan's Hardware has budgeted $67,000 to the realization of this project.

### Reliability

A failure may result in a loss of data. The impact from the loss of data is very severe. To mitigate the effect from loss of user data, budget will have to be allocated to storing user data in remote backups.

The expected mean time between failures is greater than 1 year(s).

With remote backups in place, the mean time to repair for replacement of data will be less than 4 hour(s). It is more imperative to fix the underlying cause for the loss of data, which may require the temporary disabling of the software for users and uploading an emergency patch. The expected mean time to repair underlying issues is between 1 and 3 days.

### Time Constraints

Stan's Hardware expects the software product with all its features, fully-functioning and delivered no later than December 9, 2020.

# 

# Product Specifications

## Modification History

|  |  |  |
| --- | --- | --- |
| **Date** | **Modified By** | **Changes** |
| 10/14/2020 | Edward, Christopher, Michael, Jordan | * Added Use Cases, Abstract, Class Diagram |
| 10/16/2020 | Edward, Christopher, Michael | * Modified Use Cases, Class Diagram |
| 10/20/2020 | Christopher, Michael | * Added Scenarios and Diagrams * Modified Use Cases |
| 10/23/2020 | Edward, Christopher, Michael, Jordan | * Added Glossary * Modified Use Cases |
| 10/27/2020 | Edward, Christopher, Michael | * Added User Interface * Modified Abstract, Use Cases, Glossary |
| 10/30/2020 | Christopher, Michael | * Modified Scenarios and Diagrams, Class Diagram |
| 10/31/2020 | Michael, Christopher, Edward | * Added Others * Modified Use Cases, Scenarios and Diagrams |

## Major Milestones

|  |  |
| --- | --- |
| **Date** | **Milestone** |
| 10/20/20 | Filled out the Requirements Document |
| 10/23/20 | Filled out the Software Project Management Plan |
| 10/27/20 | Implemented preliminary software |
| 10/30/20 | Completed Test Plan |
| 10/31/20 | Completed GUI and Preliminary Detailed Design |

## Abstract

Our small firm has been contracted to develop a software application to organize and maintain a database for Stan’s Hardware. The application should maintain a database with remote backup capabilities and be capable of protecting data from misuse. Users should be able to perform lookups of items and reviews and make updates to the database based on that user’s permissions. Our team of four consisting of Edward, Michael, Christopher, and Jordan will perform requirements determination, analysis, design, and testing. This application has an expected delivery date of December 9, 2020.

## Document References

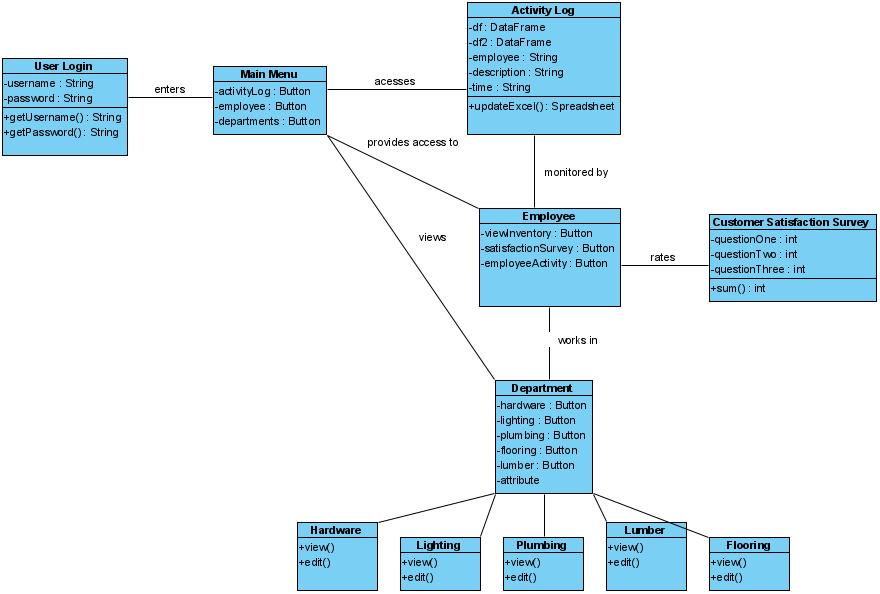
|  |  |
| --- | --- |
| **Document** | **Location** |
| Tkinter Documentation | https://docs.python.org/3/library/tkinter.html |
| Learnpython Documentation | learnpython.org |
| Object-Oriented and Classical Software Engineering 8th ed | https://www.mheducation.com/highered/product/object-oriented-classical-software-engineering-schach/M9780073376189.html |

## Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Void | To cancel a clients order. |
| Edit | To change a clients order. |
| Markdown | A reduction in price. |
| Stock | The goods or merchandise kept on the premises of a business or warehouse and available for sale or distribution. |
| Order | The summary of items a client ordered. |
| Log | An official record of events |

|  |  |
| --- | --- |
| Acronym | Meaning |
| POS (Point of Sale) | The point of sale or point of purchase is the time and place where a retail transaction is completed. |
| CRO (Conversion Rate Optimization) | Conversion rate optimization is a set of tactics ecommerce companies use to improve the ratio between site visits and conversions (downloading their apps, signing up for newsletters). |
| SH | Stan’s Hardware |
| DOB | Date of Birth |

## Class Diagram



## 

## Use Case Sequence Diagram

### 

### Use Case 1

**Actor**: Admin

**Description**:Admin user wants to add a new Supervisor, “Jane,” to the database and give her a unique ID.

**Scenario**:

1. User logs in as Admin

2. Admin presses the “Supervisor” button

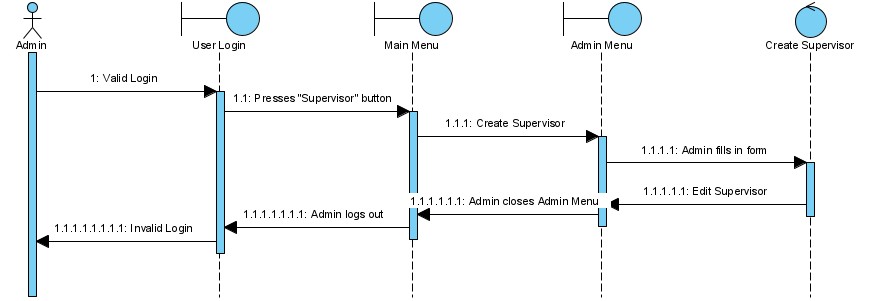
3. Admin presses the “Create” button

4. Admin fills in the form with Jane’s full name, address, and phone number.

5. Admin clicks “Submit”

6. Admin writes down Jane’s generated ID number to give to her.

**Sequence Diagram**:



### Use Case 2

**Actor**: Admin

**Description**:A Supervisor just changed phone companies and has a new phone number. Admin user must change the Supervisor’s information.

**Scenario**:

1. User logs in as Admin

2. Admin presses “Supervisor” button.

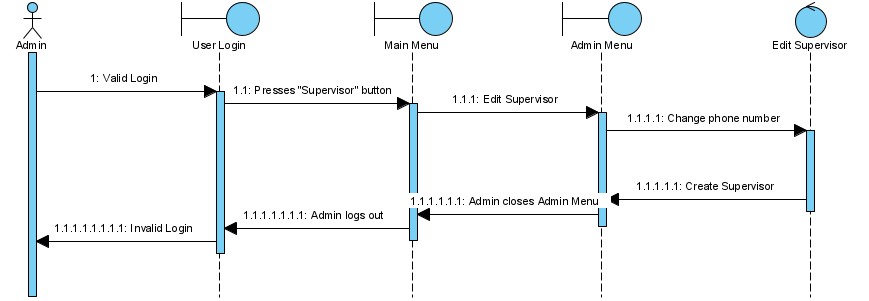
3. Admin selects the Supervisor from the drop-down menu.

4. Admin clicks “Edit” button.

5. Admin changes the phone number field of the Supervisor’s information.

6. Admin clicks “Submit.”

**Sequence Diagram**:



### Use Case 3

**Actor**: Admin

**Description**:Admin user needs to remove “John Wilkes” as he left his job 2 weeks ago..

**Scenario**:

1. User logs in as Admin.

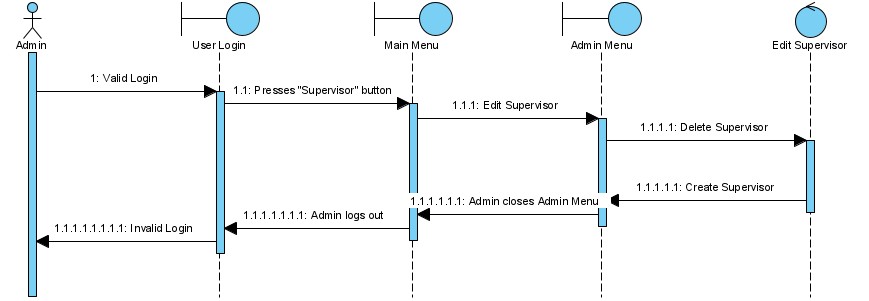
2. Admin presses “Supervisor” button.

3. Admin selects “John Wilkes” from the drop-down menu.

4. Admin clicks “Delete” button.

5. Admin clicks “Yes” when pop-up window appears.

**Sequence Diagram**:



### Use Case 4

**Actor**: Admin

**Description**:The database has not been backed up in some time and Admin needs to manually perform a backup.

**Scenario**:

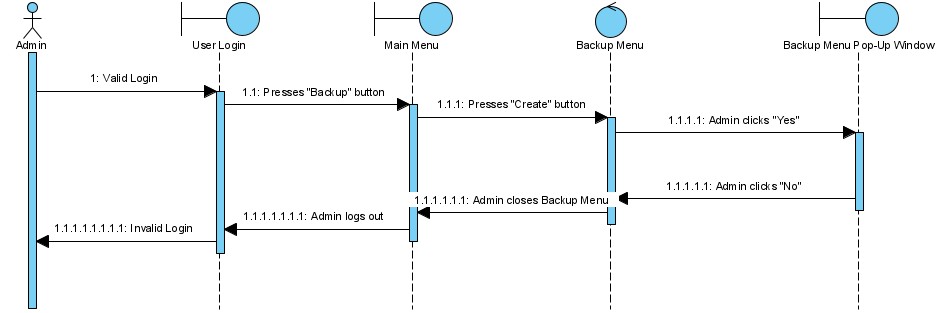
1. User logs in as Admin

2. Admin presses “Backup” button.

3. Admin presses “Create” button.

4. Admin clicks “Yes” on the pop-up window

**Sequence Diagram**:



### Use Case 5

**Actor**: Admin

**Description**:Admin notices a discrepancy in the database and must restore from yesterday’s backup.

**Scenario**:

1. User logs in as Admin

2. Admin presses “Backup” button.

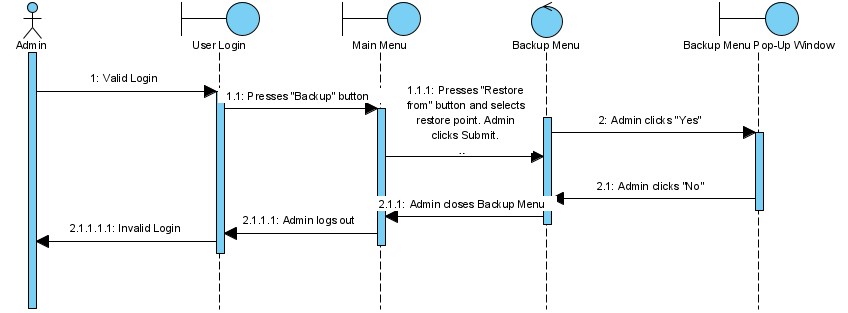
3. Admin presses “Restore from” button.

4. Admin selects yesterday’s backup from the drop-down menu.

5. Admin clicks “Submit”

6. Admin clicks “Yes” on the pop-up window.

**Sequence Diagram**:



### Use Case 6

**Actor**: Admin

**Description**:After a large personnel change, Admin has to clear the activity log of past employees.

**Scenario**:

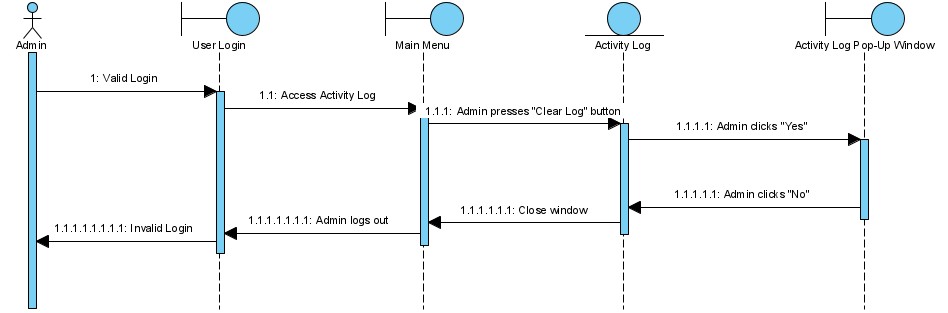
1. User logs in as Admin

2. Admin presses “Activity Log” button.

3. Admin presses “Clear Log” button.

4. Admin clicks “Yes” on the pop-up window..

**Sequence Diagram**:



### Use Case 7

**Actor**: Admin/Supervisor

**Description**:Alex has just joined the SH family and his supervisor needs to enter him into the database.

**Scenario**:

1. User logs in as Supervisor

2. Supervisor presses the “Employee” button

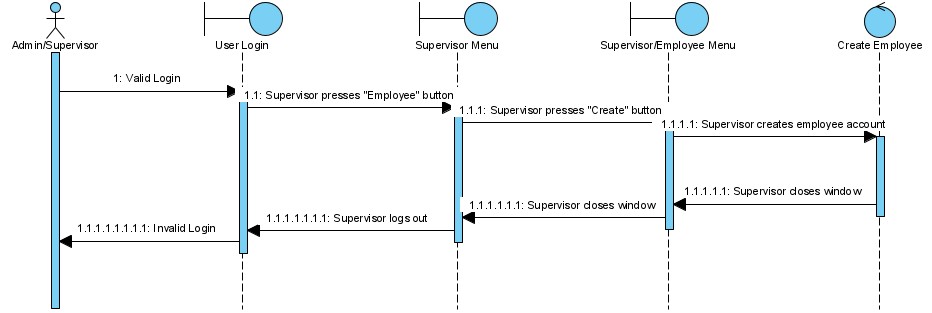
3. Supervisor presses the “Create” button

4. Supervisor fills in Alex’s phone number, street address, and full name.

5. Supervisor clicks “Submit”

6. Supervisor writes down Alex’s ID number to give to him.

**Sequence Diagram**:



### Use Case 8

**Actor**: Admin/Supervisor

**Description**:Employee “Jane” is moving in with her boyfriend and has a new street address. Her supervisor needs to update her information.

**Scenario**:

1. User logs in as Supervisor

2. Supervisor presses “Employee” button.

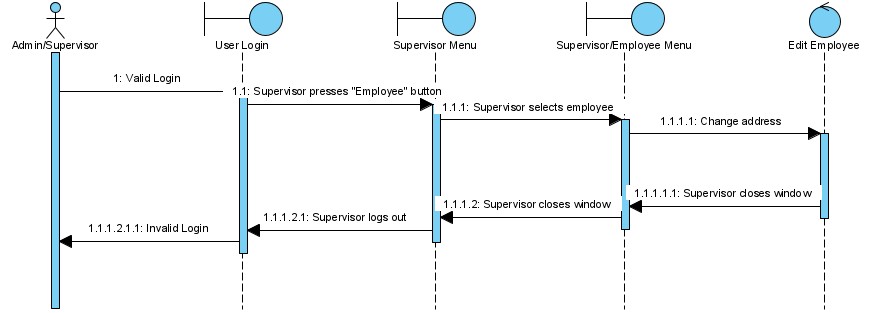
3. Supervisor selects “Jane” from the drop-down menu.

4. Supervisor clicks “Edit” button.

5. Supervisor changes Jane’s address.

6. Supervisor clicks “Submit.”

**Sequence Diagram**:



### Use Case 9

**Actor**: Admin/Supervisor

**Description**:Employee “Tom” stopped showing up for work. Admin must remove him from the database.

**Scenario**:

1. User logs in as Admin.

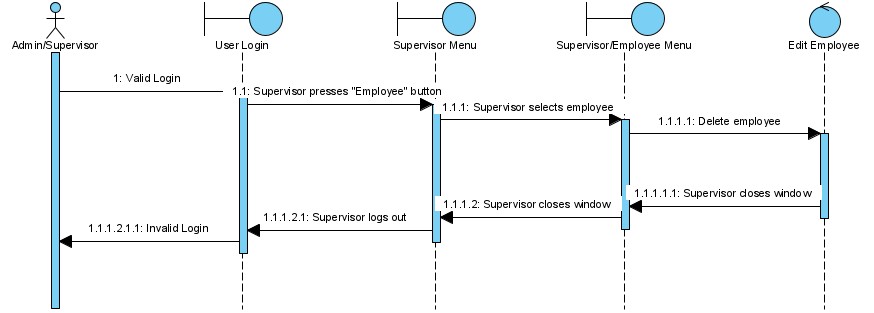
2. Admin presses “Employee” button.

3. Admin selects “Tom” from the drop-down menu.

4. Admin clicks “Delete” button.

5. Admin clicks “Yes” when pop-up window appears.

**Sequence Diagram**:



### Use Case 10

**Actor**: Admin/Supervisor

**Description**:Admin needs to do weekly reviews of the activity log for suspicious behavior.

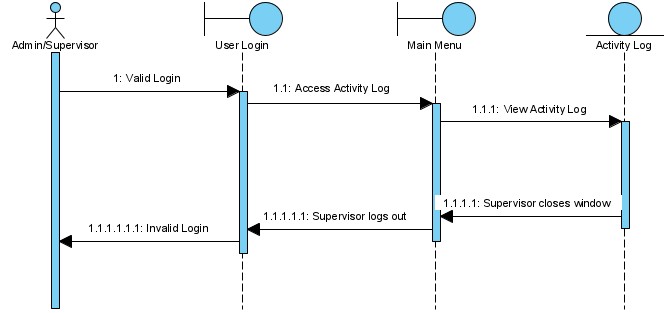
**Scenario**:

1. User logs in as Admin.

2. Admin clicks on Activity Log.

3. Admin reviews Activity Log for suspicious login times.

**Sequence Diagram**:



### Use Case 11

**Actor**: Admin/Supervisor

**Description**:Supervisor needs to make an adjustment to available hammers by removing stock. Employees are not permitted to delete inventory.

**Scenario**:

1. User logs in as Supervisor

2. (Optional) Supervisor clicks on “Department” button.

3. (Optional) Supervisor selects “Tools” department.

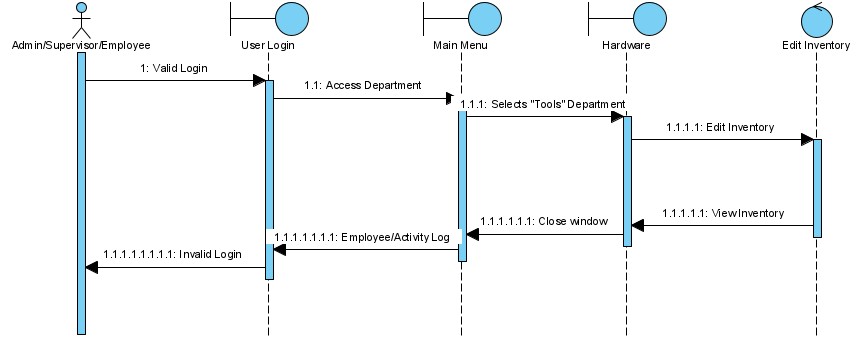
4. Supervisor enters Inventory.

5. Supervisor selects the hammer item from the menu.

6. Supervisor clicks the “Delete” button.

7. Supervisor clicks “Yes” from the pop-up menu.

**Sequence Diagram**:



### Use Case 12

**Actor**: Admin/Supervisor/Employee

**Description**:Employee “Jim” needs to give a quote to a customer on a specific brand of lawn mower.

**Scenario**:

1. User logs in as Employee.

2. Employee clicks “Department” button.

3. Employee selects “Lawn Care” from drop-down menu.

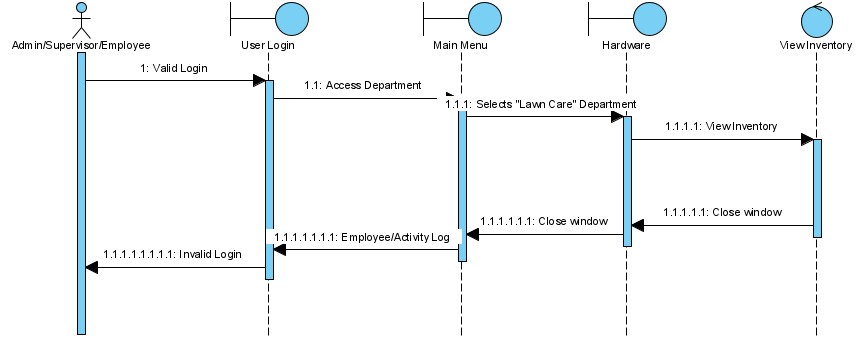
4. Employee enters Inventory.

5. Employee enters the brand into the search bar.

6. Employee selects the lawn mower from the list.

7. Employee presses “View”

**Sequence Diagram**:



### Use Case 13

**Actor**: Admin/Supervisor/Employee

**Description**:After making a sale manually, Supervisor “Tod” needs to update the inventory with less stock.

**Scenario**:

1. User logs in as Employee.

2. Supervisor presses “Inventory” button.

3. Supervisor enters the inventory ID.

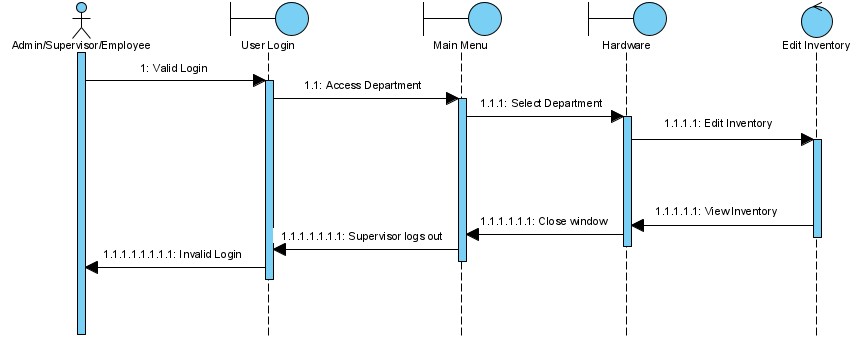
4. Supervisor selects the item.

5. Supervisor presses “Update”

5. Supervisor lowers the number in field: “Quantity”.

6. Supervisor presses “Submit” button.

**Sequence Diagram**:



### Use Case 14

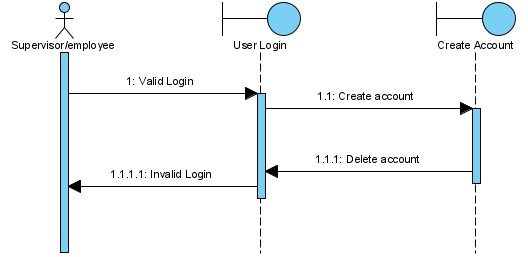
**Actor**: Supervisor/Employee

**Description**:A new hire, “Jim,” needs to use his unique ID to create a new password to log in to the system.

**Scenario**:

1. User clicks “Create Account” button.
2. User enters their ID number.
3. User fills in password form.
4. User clicks “Create”.

**Sequence Diagram**:



### Use Case 15

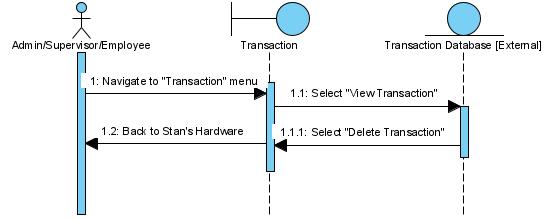
**Actor**: Admin/Supervisor/Employee

**Description**:Supervisor “Tod” has been asked to see if someone purchased 10 screwdrivers yesterday..

**Scenario**:

1. User logs in as Supervisor.
2. Supervisor clicks “Transactions” button.
3. Supervisor looks for a transaction with “Quantity” 10 and name “Screwdriver”

**Sequence Diagram**:



### Use Case 16

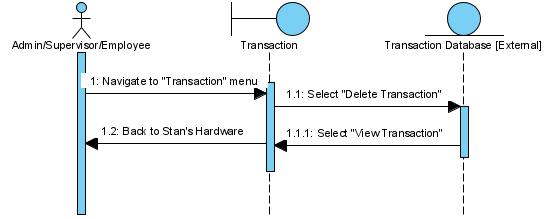
**Actor**: Admin/Supervisor

**Description**:An error has occurred with the POS machine and a voided transaction was not removed from the database.

**Scenario**:

1. User logs in as Admin.
2. Admin clicks “Transactions” button.
3. Admin chooses the transaction that was voided to delete.
4. Admin clicks “Delete Transaction”.
5. Admin clicks “Yes.”

**Sequence Diagram**:



### Use Case 17

**Actor**: Supervisor/Employee

**Description**:Employee “Jim” has been told by his Supervisor to ask every customer to rate the service they’ve been given at the end of every interaction that resulted in a transaction.

**Scenario**:

1. User logs in as Employee.

2. Employee clicks “Ratings” button.

3. Employee enters a transaction ID.

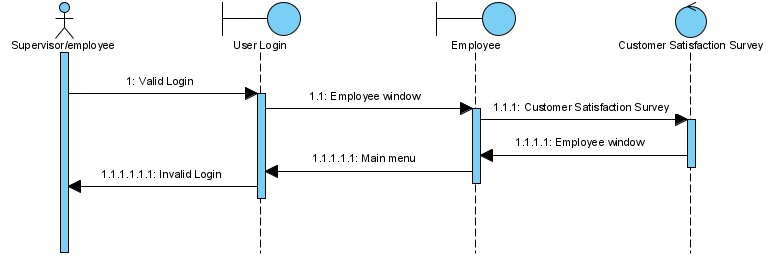
4. Employee asks the customer to rate their service.

5. Employee enters the rating.

6. Employee adds his own comments.

7. Employee presses “Submit”.

**Sequence Diagram**:



### Use Case 18

**Actor**: Admin

**Description**:An employee was giving herself false ratings. Admin needs to remove these ratings from the database.

**Scenario**:

1. User logs in as Admin.

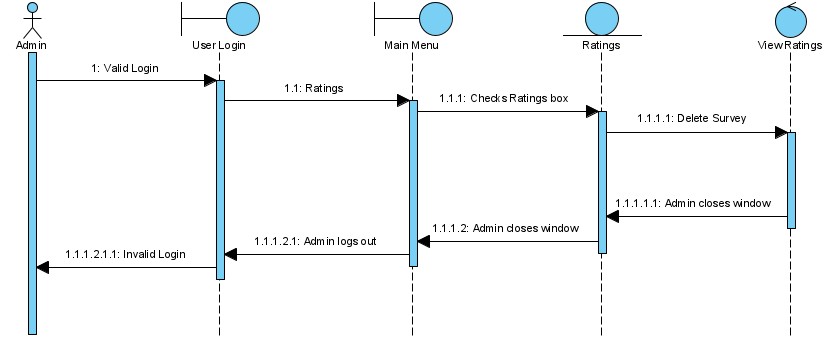
2. Admin clicks “Ratings” button.

3. Admin clicks the checkbox next to ratings to remove.

4. Admin presses “Delete” button.

5. Admin clicks “Yes” on the pop-up window.

**Sequence Diagram**:



### Use Case 19

**Actor**: Admin/Supervisor

**Description**:Supervisor “Tod” wants to see how an employee “Jim” is being rated.

**Scenario**:

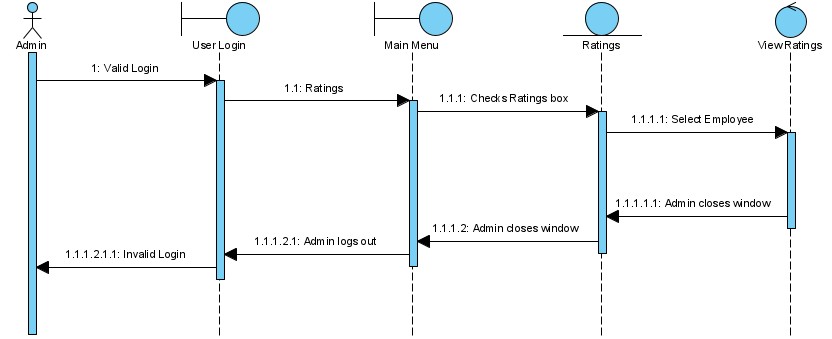
1. User logs in as Supervisor.

2. Supervisor clicks “Ratings” button.

3. Supervisor clicks on “Name” tab to sort list by name.

4. Supervisor searches for Jim’s ratings.

**Sequence Diagram**:



### Use Case 20

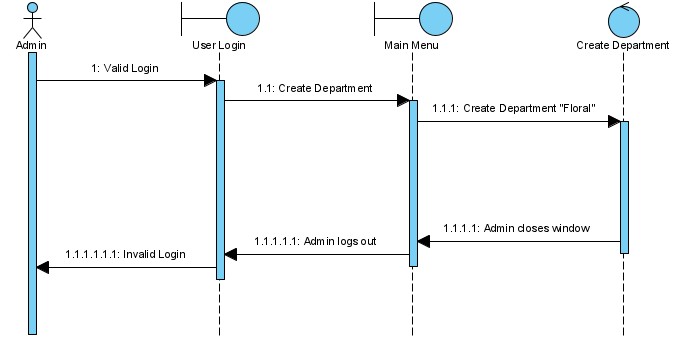
**Actor**: Admin

**Description**:SH is expanding and just added a new floral section to their store. Admin needs to add a new department to the database.

**Scenario**:

1. User logs in as Admin
2. Admin clicks on “Department button.
3. Admin clicks on “Create” button.
4. Admin enters “Floral” into the name field.
5. Admin clicks “Done” button.

**Sequence Diagram**:



### Use Case 21

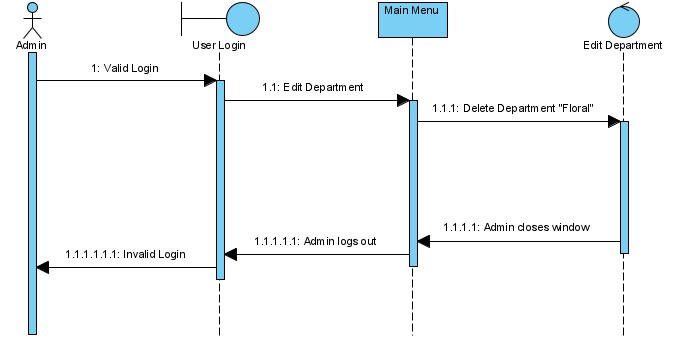
**Actor**: Admin

**Description**:SH’s floral department failed to sell a single plant and Stan has decided to remove the department from the store altogether.

**Scenario**:

1. User logs in as Admin.
2. Admin clicks on “Department” button.
3. Admin selects “Floral” from department drop-down list.
4. Admin clicks “Delete” button.
5. Admin presses “Yes” button from pop-out window.

**Sequence Diagram**:

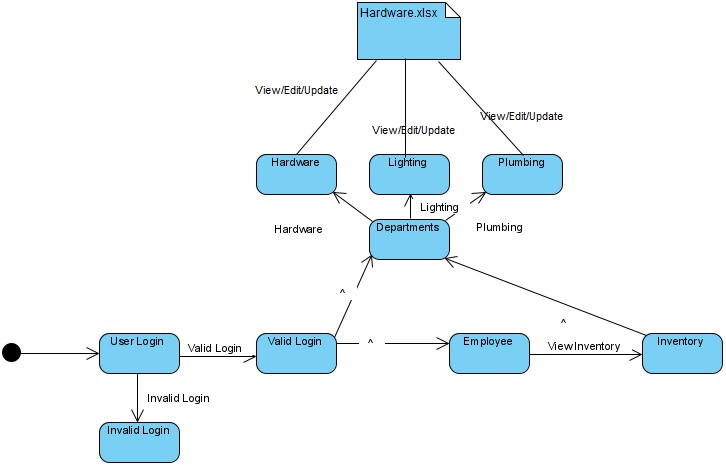


### 

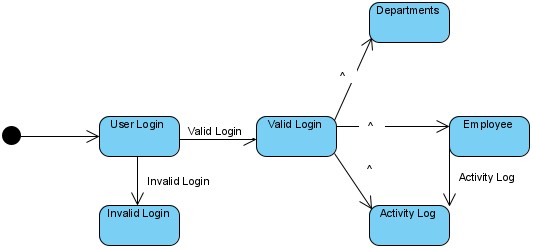
### 

***State Transition Diagrams***

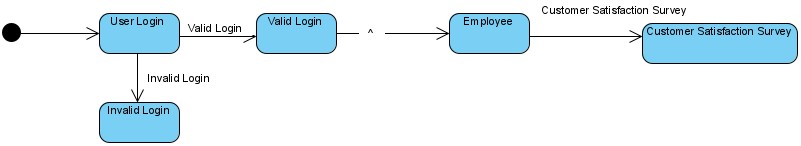
Inventory:



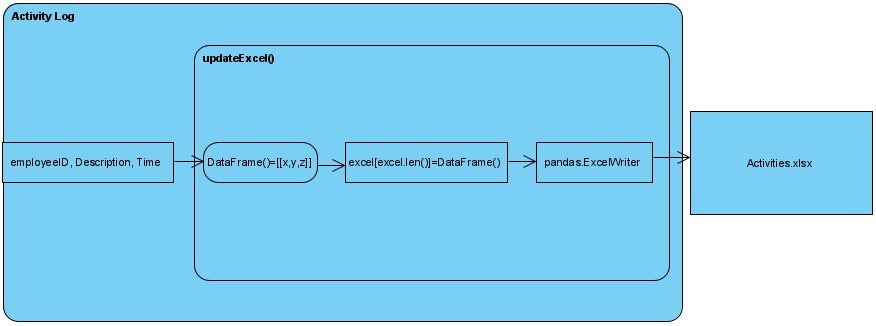
Main Menu:



Rating:

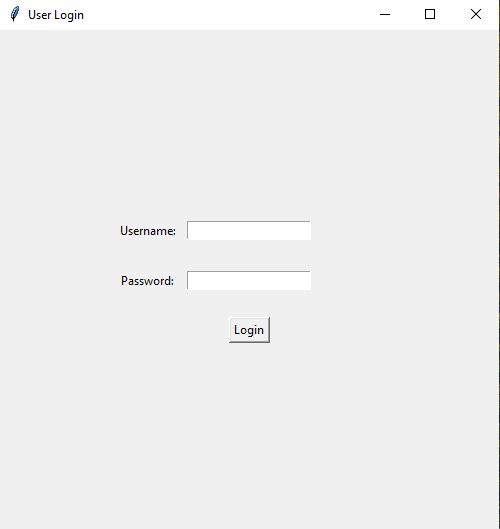


ExcelWriter:



## User Interface

### Picture/Description



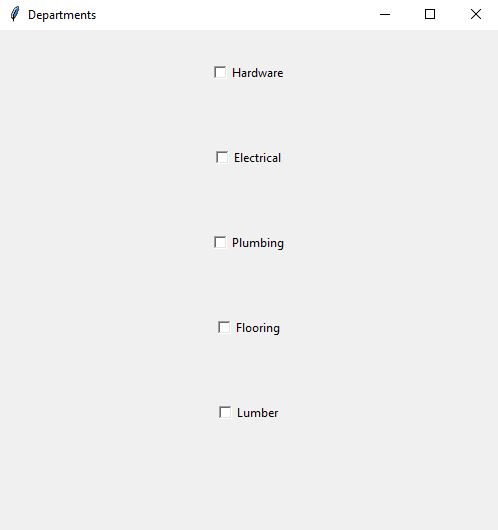
**Fig 1.1****User Login screen** *This screen accepts a username and password. If the two are a match, then the user gains admittance to the app.*



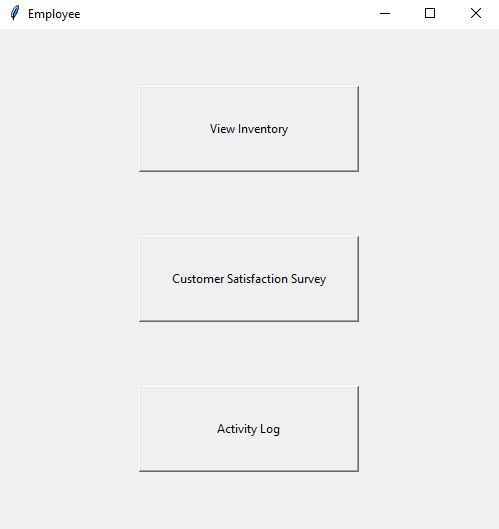
**Fig 1.2 Main Menu** *The main menu has three options: Employee, Departments, and Activity Log.*

# 

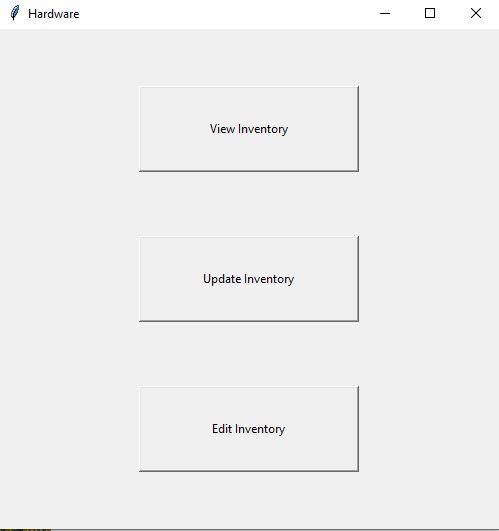
**Fig 1.3 Activity Log** *The activity log is how employees, supervisors, and managers keep track of daily activities. All entries are logged in an Excel spreadsheet.*



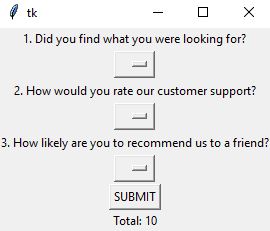
**Fig 1.4 Departments** *There are five departments: Hardware, Lighting, Plumbing, Flooring, and Lumber.*



**Fig 1.5 Employee** *Employees can view the inventory of any department, the contents of the activity log, or access the prompt for the Customer Satisfaction Survey.*



**Fig 1.6 Department Screen** *There are three identical department screens, one for each department. Employees can view the inventory. Managers can edit inventory.*



**Fig 1.7 Customer Satisfaction Survey** *The Customer Satisfaction Survey rates an employee’s performance on a scale of 0-15.*

## Others

|  |  |
| --- | --- |
| **Description** | **Link or Reference** |
| Preliminary Sketch of User Interface | https://imgur.com/a/OGhqJiZ |

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Software Project Management Plan

## PREFACE

## 1. Overview of the Product

### Purpose Scope & Objective

The purpose of the product is to provide a software solution for the employees and

managerial staff of a Stan’s Hardware.

### 1.1 Project Background

This project is to provide a software to manage incoming orders. Employees and managers will be able to log into the system and check inquiries on supplies ordered. The most significant use is managers being able to see employees activities and ensuring everything is in order. The system will have a simple GUI for users to interact with, the database will be pre-implemented, and software will be simply ran on a windows machine.

### 1.2 Assumptions & Constraints

We assume the user has no knowledge of database management and very little computer knowledge. The software is intended to run on 6 Dell desktop computers running Windows 10, interfacing with an offsite server with data storage and cloud backup capabilities. The developers of the software are Michael Giraldi, Christopher Pando, Edward Aranda, and Jordan Adriano. The project is due December 9, 2020 and has an overall budget of $67,000.

### 1.3 Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **Major Deliverable** | **Planned Delivery Date** | **Author** | **Delivery Mechanism** |
| User Manual | 12/09/2020 | Edward Aranda | Paper and PDF |
| Inventory Interface | 12/09/2020 | Christopher Pando | CD-ROM |
| Offsite SQL Server | 12/09/2020 | Michael Giraldi | Connection via cloud |
| Customer Satisfaction Interface | 12/09/2020 | Jordan Adriano | CD-ROM |

### 1.4 Schedule & Budget Summary

The workflow are as follows:

Requirements workflow (1 week, 3 team members, $4,500)

Analysis workflow (1 week, 3 team members, $5,800)

Design workflow (2 weeks, 4 team members, $10,800)

Implementation workflow (2 weeks, 4 team members, $12,300)

Testing workflow (2 weeks, 3 team members, $8,200)

Total development time is 8 weeks, four team members, with an internal development cost of $41,600.

### 1.5 Evolution of the Plan

All changes in the project management plan must be approved by Edward before they are implemented. All changes should be documented to keep the project management plan correct and up to date.

# 

# 

## 2. References

|  |  |  |
| --- | --- | --- |
| **Resource** | **Identifier** | **Description of Use** |
| Object-Oriented and Classical Software Engineering 8th ed | The McGraw-Hill Companies, Inc. | Used as a guideline on how to fill out documents 1-3 |
| Learnpython.org  (Website) | Learnpython.org | Training Tool |
| https://docs.python.org/3/library/tkinter.html  (Website) | Tkinter | Training Tool  Documentation |
| https://www.google.com/docs/about/  (Website) | Google Docs | Training Tool  Documentation |
| https://zoom.us/  (Website) | Zoom | Communication Tool |

## 3. Definitions & Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Void | To cancel a clients order. |
| Edit | To change a clients order. |
| Markdown | A reduction in price. |
| Stock | The goods or merchandise kept on the premises of a business or warehouse and available for sale or distribution. |
| Order | The summary of items a client ordered. |
| Log | An official record of events |

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| POS (Point of Sale) | The point of sale or point of purchase is the time and place where a retail transaction is completed. |
| CRO (Conversion Rate Optimization) | Conversion rate optimization is a set of tactics ecommerce companies use to improve the ratio between site visits and conversions (downloading their apps, signing up for newsletters). |
| SH | Stan’s Hardware |
| DOB | Date of Birth |

## 4. Project Organization

### 4.1 External Interfaces

All the work on this project will be performed by Edward, Michael, Christopher, and Jordan. Edward and Michael will meet every week to deliver and discuss that week’s project progress report with the client.

### 4.2 Internal Structure

The development team consists of Jordan, Christopher, Michael, and Edward

### 4.3 Roles & Responsibilities

Michael, Jordan, and Edward will develop the requirements workflow. Christopher and Edward will begin development of the implementation. Jordan and Edward will construct the artifacts handling documentation. Michael will develop the artifacts handling operation expenses and development costs. Edward will oversee all phases of the project’s workflows.

|  |  |  |
| --- | --- | --- |
| **Roles** | **Responsibilities** | **Person** |
| Project Leader | Ensuring all requirements are met. Also in charge of scheduling meetings and deadlines. | Edward Aranda |
| Secretary | Oversees code development and coordinates technical aspectects of the project. | Jordan Adriano |
| Secretary | Oversees code development and develops the source code. Instructor to new libraries. | Michael Giraldi |
| SQA Representative | Head developer of the source code. Implements new libraries and updates source code. | Christopher Pando |

### 4.4 Team Methods of Operation

Edward will call meetings on Zoom every Tuesday and Friday to collaborate. In these Zoom calls, team members will describe the status of their portion of the project, their tasks to be completed, and any issues they need assistance with. Meetings will be coordinated through GroupMe. Any concerns or discussions outside of meeting hours will also be handled through GroupMe.

## 5. Managerial Process Plans

### 5.1 Start-Up Plan

The project effort, cost, and schedule will be estimated by how much work is done within each zoom meeting. Team members will meet up multiple days a week to work on the project. The zoom meetings have a set day and time which does not change. The cost of the project is estimated by how many hours an employee (us) worked on the project.

The project effort, cost, and schedule will be estimated by how much work is done within each zoom meeting. Team members will meet up multiple days a week to work on the project. The zoom meetings have a set day and time which does not change. The cost of the project is estimated by how many hours an employee (us) worked on the project.

Staffing is managed according to people's availability and set skills. Each person's schedule is limited and must be accounted for while scheduling them to progress the project. Each staff member is required to know the basics of using the windows operating system and know the fundamentals of programming. Furthermore, each staff member must be equipped with the basics of using python and its respected IDE. We also need the staff to source essential libraries found within python.

### 

### Training

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Training** | **Method** | **Scheduled Dates** | **Cost** |
| OpenEDG Python Institute | Certified Entry-Level Python Programmer Certification | 10/08/20-11/19/2020 | $0 |
| Databases | UHD course | Previously taken | $700 |
| Programming 1 and  Programming 2 | UHD course | Previously taken | $700 |

### 5.2 Work Plan

#### 5.2.1 Work Activities

The work and activities that are achieved by the staff are recorded and documented in the milestones page in relationship to work breakdown and date. Members will update on individual progression throughout the project at the start of each meeting. If any issues or problems arise, that would be the moment to make them known. Team members will also meet at the end of each meeting and discuss problems and progress. Meetings that are held on Friday to report progress and determine if any changes need to be made. Edward will ensure that schedule and budget requirements are met.

Specify (or refer to a location that contains a list of) the work activities and their relationships depicted in a work breakdown structure. Decompose the structure to a low enough level to facilitate sound estimating, tracking, and risk management. Activities and tasks detailing the approach, needed resources, duration, work products, acceptance criteria, predecessors and successors.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Project Member** | **Phases Involved** |
| Project Planning | Edward Aranda | Planning Phase  Requirements Phase  Design Phase |
| Scope management | Jordan Adriano  Michael Giraldi | Requirements Phase  Analysis Phase  Implementation Phase |
| Project estimation | Michael Giraldi | Implementation Phase |
| Design | Edward Aranda | Planning Phase  Requirements Phase  Design Phase |
| Testing | Christopher Pando | Analysis Phase  Implementation Phase |
| Maintenance | Jordan Adriano | Analysis Phase  Transition Phase  Documentation Phase |

#### 

#### 5.2.2 Schedule Allocation

Week 1. (Completed) Held a meeting to determine project and determined requirements artifacts. Inspected requirements artifacts.

Weeks 2-3. (Completed) Produced analysis artifacts, and inspected analysis artifacts. Shared artifact with team leader (Edward), who approved them. Produced software project management plan, and inspected software project management plan.

Weeks 4-5. (Completed) creation of product designs, reviewing product designs, and documentation of product designs.

Weeks 6–10. Creation and inspection of each class, reviewing and updating documentation, integration of each class, GUI source code development, database maintenance, create and modify employee spreadsheet, create and log employee activity, create and log invoices, create an admin version control.

#### 

#### 5.2.3 Resource Allocation

The application will require 3 months and 12 person-months to complete.

Week 1. (Completed) 3 hours and 12 person-hours. We staffed the office and trained new hires. Initial drafting and design. $4,000 were spent on salary and training.

Weeks 2,3. (Completed)6 hours and 24 person-hours. We performed project analysis and shared documents with the team leader. Further drafting and design, transition to PyCharm. $8,000 were spent on salary and software tools.

Weeks 4,5. (Completed)11 hours and 44 person-hours. We are filling out documents and drafting the initial application. $8,000 were spent on salary and legal fees.

Weeks 6-10. We are entering the final stages of coding and drafting the GUI. Documentation and major prototypes to be completed and submitted to team leader for approval.

#### 

#### 5.2.4 Budget Allocation

Requirements workflow $10,000

Analysis workflow $3,200

Design workflow $3,200

Implementation workflow $25,000

Testing workflow $12,000

Total $53,400

### 5.3 Control Plan

The client’s needs have changed and required the team to start a new project. The milestones and budget have been affected significantly since the previous project was almost fully developed. The change was approved by team leader, Edward, and will be responsible for ensuring that the project is completed on time and within budget. This requirement can be met by hosting daily zoom meetings with team members. At each meeting team members; Jordan, Michael, Christopher, Edward present the progress being made using the specification document and the project management plan. Any major problems faced by the team members will immediately be reported to Edward.

#### 5.3.1 Requirements Management

Describes the method to be used to evaluate, report, and propose changes to the project as needed.

|  |  |  |
| --- | --- | --- |
| **Requirements Management Activities** | **Performed By Whom** | **Comments** |
| Establishing the requirements | Edward Aranda | Issues the necessary requirements needed throughout the project |
| Review and evaluate requirements change request | Edward Aranda  Christopher Pando | Reviews all proposed changes throughout the project. |
| Provides impact analysis for proposed changes | Jordan Adriano  Michael Giraldi | Provides insights on how the changes will alter the project in its entirety. |
| Notifier of approved changes | Edward Aranda | Notifies the entire team on any approved changes to be made throughout the project. |
| Incorporate requirements into project plan | All Staff | Staff involved to make sure approved changes are incorporated throughout the project. |
| Evaluator of risk due to approved changes | Christopher Pando  Michael Giraldi | Evaluates the new risk changes could cause throughout the project. |
| Evaluates changes in the planned cost, schedule, or quality. | Edward Aranda | Evaluates how changes are going to affect the cost or quality. In addition, will adapt the scheduling to new changes. |

#### 

#### 5.3.2 Schedule Control

**Inputs:** project management plan, project schedule, work performance data ,project calendar, schedule data.

**Tools and techniques:** performance review, project management software, resource optimization techniques, modeling techniques, leads and lags

**Output:** work performance information, schedule forecast, change request, project management plan updates, project document updates

#### 5.3.3 Resource Control

Resources are allocated and controlled by individuals assigned tasks within the set resource. Zoom meetings resources are set and controlled by the team leader, Edward Aranda, to ensure the meetings are available to all staff members. Google docs resources are controlled by Michael Giraldi to ensure all members can edit at the same time. In addition, this ensures quality of the documentation. Lastly, Christopher Pando has complete control of the source code on python to also ensure quality. The source code is then distrubed during the meeting to ensure everybody is on the same version.

#### 5.3.4 Budget Control

Performance budget can be monitored and controlled by removing access to duplicate software that UHD grants access to. Costs will be tracked by making sure all members have access to the required software that UHD offers without paying it on their own. The cost of the project will be $53,400 while contract labor and support functions will make that cost go up.

#### 5.3.5 Reporting & Communication Plan

Documentation is contained within a Google Doc that is hosted by Michael Giraldi to be edited by all staff members. A zoom meeting is held to communicate with each member as documentation is updated. All issues or concerns are brought up in real time and can be resolved quickly and efficiently.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Information** | **Frequency** | **From** | **To** | **Medium** |
| Documentation | Weekly | 1:30PM | 3:30PM | Google Drive |
| Progress | Daily | 1:30PM | 5:00PM | Zoom |
| Scheduling | Daily | 8:00AM | 5:00PM | GroupMe |
| Diagrams | Daily | 8:00AM | 5:00PM | GroupMe |
| Source Code | Daily | 8:00AM | 5:00PM | GroupMe |

#### 5.3.6 Measurement Plan

Details about the projects measures that will be selected. This consists of project team effort, key issues faced by the project, and external requirements. Measured requirements, Analyzed, Reports, and uses are being evaluated from a business perspective.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Measure Required** | **Frequency Collected** | **Collected By Whom** | **Analyzed By Whom** | **Used By Whom** |
| Size Related Metrics | Weekly | Christopher Pando | Edward Aranda | Stanley Strange (Client) |
| Function Related Metrics | Weekly | Jordan Adriano  Michael Giraldi | Edward Aranda | Stanley Strange (Client) |
| Cycle Time | Bi-weekly | Michael Giraldi | Edward Aranda | Stanley Strange (Client) |
| Lead Time | Bi-weekly | Christopher Pando | Edward Aranda | Stanley Strange (Client) |

## 

### 5.4 Risk Management Plan

A detailed plan that will be used to identify, analyze the impact, build, plan, and manage the risks involving the project.

|  |  |  |
| --- | --- | --- |
| **Risk Management Activity** | **Performed By Whom** | **Comments** |
| Possible risk penalty observation | Edward Aranda | Estimate for each risk item the probability of occurrence and loss if the risk occurs; compute exposure and weigh out the options. |
| Risk List Update | Edward Aranda | Periodically analyze the project status to identify new risks as they occur. |
| Staff Risk Evaluation | Edward Aranda | Evaluating the risks of each staff member's involvement from staff changes/rotation. |
| Risk Ranking | All Head Staff | Ranking the severity of each risk and organizing them |
| Budget Risks | Edward Aranda | Evaluating the budget management risks |

### Risk Management Note

All Risk Management Activity must be reviewed and approved by Edward Aranda. Note that all Risk Activities are subject to change.

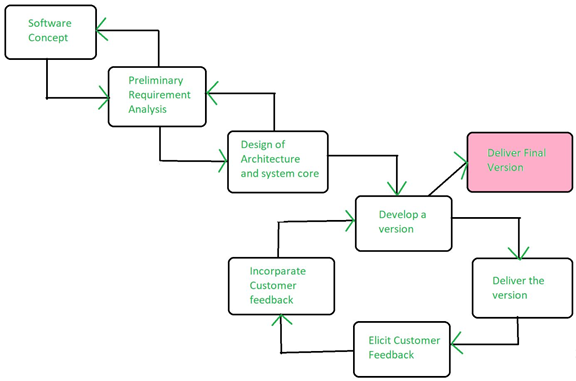
### 5.5 Closeout Plan

|  |  |  |
| --- | --- | --- |
| **Closeout Activity** | **Performed By Whom** | **Comments** |
| Product testing | Chris Pando | Test for variation of user interaction. Debug code. |
| Verify database backups | Edward Aranda | Ensure that the database can be reconstructed from backup. |
| Verify documentation | Jordan Adriano | Check for errors. |
| Compile customer satisfaction surveys | Michael Giraldi | Employee performance analysis. |

## 6. Technical Process Plans

### 6.1 Process Model

The Waterfall sequence diagram consists of breaking down the project into linear phases that are linear, where each phase depends on the last in order to progress.



### 6.2 Methods, Tools, & Techniques

#### Programming Environment

Python programming language will be used to implement the diagram/overview of the project

#### Database Environment

The current project does not require a database to function. Spreadsheet will be used instead to keep records of information and inventory.

#### Version Control

azure devops version control will be used to control the database we choose in future development.

#### Documentation

Google Docs is used by all staff members to edit the documentation at the same time.

#### Testing

All testing is done on staff’s personal computers and will continue to be tested within the staff team.

### 6.3 Infrastructure Plan

The software will be installed on 6 terminals running Windows 10. It requires at minimum an Intel i3/AMD Ryzen 3 processor, 4 GB of RAM and 600 MB of disk space.

### 

### 6.4 Product Acceptance Plan

The final product will be delivered to the customer on December 9, 2020.

The application will provide a secure login experience.

The application will build and maintain a database in Microsoft Excel

The application will accommodate 23 employees, 5 managers, and an administrator.

Acceptance of the product by our client will be achieved by following the steps of the Unified process.

### 

### 6.5 Deployment Plan

The product will be contained in a CD and should be installed through a CD Rom reader. The clients will need to have a CD Rom or CD reader adapter installed to extract the software contained within the CD. Once the CD is extracted, the user will run the set up guide and follow the instructions.

All products are shipped across by land after order is confirmed. The Client should expect a 5 to 7 business day wait. All shipping methods are hosted by FedEx transportation.

## 7. Supporting Process Plans

### 7.1 Configuration Management Plan

Features will be added or removed on an as needed basis to be determined by the client. Audits will be performed on a weekly or bi-weekly basis as set forth in the Risk Management plan.

### 7.2 Product Testing & Reviews Plan

The products will be tested using Pycharm IDE, and code will have technical review by the other members (Team Leader, SQA Rep) and will be conditionally reviewed in Visual Studio. Testing will be done from various user perspectives, including but not limited to employee, manager, and administrator. Tests will be planned by the team leader, developed and carried out by the secretary, and reviewed by the SQA representative.

### 7.3 Document & Work Product Plan

**Requirements**

Problems that need to be solved are python coding for GUI, creating a backup that restores all files properly, adjusting settings for different kind of user (employee,manager,admin)

**Product Specification**

The final product will require a PC with a CD Rom, the PC will need to have windows OS and python installed before reading the CD. The python code will be stored in the CD and will use a TKinter as the GUI interface. Examples of design constraint could be the lack of PC with CD Rom or running on the wrong OS.

**Design Documentation**

The product will be designed using a bottom-up architecture. Necessary external dependencies will be imported from the internet. User interfaces will include a GUI made in Tkinter. Databases will be managed in Microsoft Excel. Internal interfaces will include a user login screen with password security.

**Implementation Documentation**

The product will be installed on 6 terminals running Microsoft Windows 10. Upon implementation, it will support a staff of 23 employees, 5 managers, and 1 administrator. The product will be installed via CD-ROM.

**Test Documentation**

Test plans that will be used for the workflow of the project will be the documentation. Test procedures will be tested by comparing the “Project Diagram” and “cases” and check if python code meets functionality and requirements. Test cases will be a step by step process which requires the team members writing the code check if the code meets the final goal (pass or fail). As of now only Alpha testing has been implemented and will be used to correct bugs in the future.

### 7.4 Quality Assurance Plan

Edward and Christopher will test each other's code, and Jordan will conduct integration testing. Product testing will be performed by all the members of the group.

### 

### 7.5 Project Progress Reviews

Project progress reviews will be conducted by the SQA representative on a bi-weekly basis as proposed in the Measurement plan. The product is to be tested in accordance with its use cases and for use variance.

### 7.6 Issue Management

Specific issues are delegated by Edward to be handled by an individual team member as listed below.

|  |  |  |
| --- | --- | --- |
| **Issue Management Activities** | **Performed By Whom** | **Comments** |
| Submit Issues | Michael | Any problems faced will be submitted for review |
| Record Issues | Jordan | Submitted issues will be recorded |
| Assign Issues | Christopher | Christopher will assign which issues will be resolved by whom |
| Track and Review Issues | Edward | Team leader will be up to date and prioritize main issues over small ones |
| Update Issues Log | Edward | From the list of issues the main issues will be kept, others may be discarded. |
| Report Status on Issues | Edward | Team members will receive update on the issues that have not been resolved |

### 7.7 Version Management

|  |  |  |
| --- | --- | --- |
| **Change Management Activities** | **Performed By Whom** | **Comments** |
| Change Request | Jordan Adriano | Submit change request describing the proposed change and indicating the importance to the user |
| Request Importance | Christopher Pando | Determine which requests have more significance and arrange them in a list. |
| Request Evaluation | Michael Giraldi | Review and evaluate the top changes according to list |
| Project Estimation Update | Edward Aranda | Estimate the required work effort and the impact of the change to the project schedule, cost, and resource availability |
| Change Approval | Edward Aranda | If change is major and does not overdue funding, resources, or schedule it will be approved by team leader (Edward) |

### 

### 7.8 Subcontract Management (Acquisition Management) Plan

Subcontracted work will be contracted by the project manager on an as needed basis. The work to be performed will be determined by necessity and additional team members will be selected by the project manager and subject to approval by the administrator. Any additional hires will be subcontracted by the secretary for a period of at least 2 weeks and not to exceed 4 weeks.

### 7.9 Process Improvement Plan

The processes of the project will be managed by the data flow diagram. This diagram will be updated on an as needed basis by the software team on an as needed basis, to be determined by the project manager. Quality of work will be assessed by the SQA representative on a weekly basis; productivity and efficiency will be evaluated by the project manager on a bi-weekly basis as stated in the measurement plan.

## Document Control

### Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description (Including Page #’s)** |
| 1.0 | 9/1/20 | Edward, Christopher, Michael, Jordan | Python code and GUI begin getting created. |
| 1.1 | 10/1/20 | Edward, Christopher, Michael, Jordan | Use cases and documentation begin. GUI and code continues getting updated. Code transitions to Python and GUI is made using Tkinter. |
|  |  |  |  |
| 1.2 | 10/15/20 | Edward, Christopher, Michael, Jordan | (Project: Stans Hardware POS System)  Brainstorming and designing the software/diagram takes place for new project |
| 2.0 | 10/20/20 | Michael, Jordan | Documentation begins to get filled out, as well as implementing new sequence diagrams, and use cases. |
| 2.1 | 10/23/20 | Edward, Christopher | The GUI and code continue being written in python according to the use cases and diagrams shown in the Requirements document. |
| 2.2 | 10/25/20 | Edward, Jordan, Christopher, Michael | All members of the group continue filling out documentation according to their role (SQA, team leader, etc) |
| 2.3 | 10/27/20 | Christopher, Edward | GUI and python code continues getting updated according to clients needs and with addition no new use cases. |
| 2.4 | 10/29/20 | Jordan, Michael | Documentation is completed. Team leader reviews documentation and points out possible areas of improvement. |
| 2.5 | 10/31/20 | Edward, Jordan, Christopher, Michael | Documentation gets approved by the team leader (Edward) and is ready for submission for Phase 2. |

### Document Storage

This document was created with Google Docs. Document is also saved in personal OneDrive for each member.

### Document Owner

This document will be maintained by Edward Aranda, Michael Giraldi, Christopher Pando, and Jordan Adriano. The original owner of the document is Edward, but was shared with the other members.

## Appendices

|  |  |  |
| --- | --- | --- |
| **Appendix** | **Title** | **Location Or Link** |
| Appendix A | Project Activities | <https://www.tutorialspoint.com/software_engineering/software_project_management.htm> |
| Appendix B | Software Development Cycle | <https://medium.com/@jilvanpinheiro/software-development-life-cycle-sdlc-phases-40d46afbe384> |
| Appendix C | Levels of Testing | <https://www.guru99.com/levels-of-testing.html> |

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Detailed Design

## Functions (Pseudocode)

### Activity Log

Define frame to be 500x500 pixels

Title = ‘Activity Log’

Initialize DataFrame to Excel Spreadsheet ‘Activity’

3 string: variables Employee, Description, Time

3 labels: Employee, Description, Time

3 entry forms: Employee, Description, Time

1 button: Update

Update Microsoft Excel Spreadsheet Function:

updateExcel(x,y,z):

Global count

DataFrame.loc[count] = [x,y,z]

Writer = to\_Excel.DataFrame() #Write to Excel

count+=1

### Departments

Initialize frame to be 500x500 pixels

Title = ‘Departments’

5 integer variables: op1,op2,op3,op4,op5

5 Check Buttons: Hardware, Electrical, Plumbing, Flooring, and Lumber

### Employee

Initialize frame to be 500x500 pixels

Title = ‘Employee’

customerSatisfactionSurvey():

Question 1: “Did you find what you were looking for?”

Question 2: “How would you rate our customer service?”

Question 3: “How likely are you to recommend us to a friend?”

3 integer variables: a1, a2, a3

sum(x,y,z):

return x + y+ z

Post label with sum value of question response integer variables

3 Labels: Questions 1, 2, and 3

3 Option Menus with ratings 1-5

Submit button that sums the three Option menus

### Login

clicked():

If usr[password]==password:

Open Main menu

Initialize frame to 500x500 pixels

Title ‘Main Menu’

Three Buttons: Employee, Department, and Activity Log

Else:

Error Message

### Main

\_\_main\_\_()

Define frame to 500x500

Title = ‘User Login’

Two strings = username, password

Two labels: pwd\_label, usr\_label

One button = submit

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Software Quality Assurance Plan (Test Plan)

## Document History and Distribution

### 1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision #** | **Revision Date** | **Description of Change** | **Author** |
| 1.0 | 10/27/2020 | Test items, Features, and Approach completed | Edward, Christopher, Michael |
| 1.1 | 10/30/2020 | Criteria, Processes and Requirements completed | Edward, Christopher, Michael, Jordan |
| 1.2 | 10/31/2020 | Modified scope and procedures | Edward, Michael |
| 1.3 | 10/31/2020 | Plan Approval | Edward, Michael |

### 2. Distribution

|  |  |  |
| --- | --- | --- |
| **Recipient Name** | **Recipient Organization** | **Distribution Method** |
| Mr. Stanley Strange | Stan’s Hardware | PDF by Email |

## 1. Introduction

This product is intended to replace SH’s current database implementation of modifying a shared Google Sheets document. The current implementation contains security, reliability, and ease-of-use issues stemming from multiple persons having unlimited access to the sheet. This product attempts to alleviate these issues by adding layers of security, backup, and additional functionality such as improved item lookup and GUI-based update features.

### 1.1 Objectives

The objective of this test plan is to determine the functionality of the current implementation of the product. This phase will test the 7 primary functionalities of the database and the results against expectations and the project deliverables. Our team will test the correctness of the database functionalities(update, add, delete, view) against expected results as well as test the security features to ensure users have the exact promised privileges. Testing is to be performed on at least 2 Windows laptops capable of running python scripts on a budget of $8,200. Successful testing will result in an Alpha-level product with preliminary database functions and GUI features, and constitute a major milestone.

### 1.2 Testing Strategy

Our testing strategy consists of manually testing various add, delete, update, and view functions as admin, employee, or supervisor. Testing should ensure that users without privileges are not allowed to modify the database and that the database is robust against malfunction. Testing will also ensure that features such as adding new departments or users work as intended. Testing will be conducted to determine if all buttons work as intended and updates are reflected in the database. Testing will also ensure that invalid entries do not make it to the database and all other errors are handled correctly.

### 1.3 Scope

All changes to the test plan are documented in Document History of SQA Plan. Scheduled updates to the SQA Plan will take place at the beginning of each testing phase. Unscheduled updates will take place as needed and will be accompanied by a notice via email to concerned parties. All software updates will be pushed to the company’s GitHub page.

### 1.4 Reference Material

* Yuan, Shengli. *Team Project - Phase 2*. UHD, 24 Aug. 2020, http://cms.dt.uh.edu/faculty/yuans/courses/cs3321/project/phase2.htm. Accessed 27 Oct. 2020
* Schach, Stephen R. *Object-Oriented and Classical Software Engineering*. New York, NY, The McGraw-Hill Companies, Inc, 2011.

### 1.5 Definitions and Acronyms

#### Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Void | To cancel a clients order. |
| Edit | To change a clients order. |
| Markdown | A reduction in price. |
| Stock | The goods or merchandise kept on the premises of a business or warehouse and available for sale or distribution. |
| Order | The summary of items a client ordered. |
| Log | An official record of events |

|  |  |
| --- | --- |
| Acronym | Meaning |
| POS (Point of Sale) | The point of sale or point of purchase is the time and place where a retail transaction is completed. |
| CRO (Conversion Rate Optimization) | Conversion rate optimization is a set of tactics ecommerce companies use to improve the ratio between site visits and conversions (downloading their apps, signing up for newsletters). |
| SH | Stan’s Hardware |
| DOB | Date of Birth |

## 2. Test Items

There is usually only one test object getting tested at a time, but that one object contains many test items which have their own set of tests to determine a PASS or FAIL result.

### 2.1 Program Modules

Department Module: Will perform Add and Delete actions as Admin.

Inventory Module: Will perform Add, Update, and Delete actions as Admin/Supervisor/Employee.

Activity Log Module: Will perform Add, View, and Delete actions as Admin.

Supervisor Module: Will perform Add, Update, and Delete actions as Admin.

Employee Module: Will perform Add, Update, and Delete actions as Admin and Supervisor.

Rating Module: Will perform Add, View, and Delete actions as Admin/Supervisor.

Transaction Module: Will perform View and Delete actions as Admin/Supervisor.

### 2.2 Job Control Procedures

Testing will be performed on the activity log to ensure that adequate resources are allocated to the activity log database. Testing procedures will include the updating of the log when an employee accesses the ‘Hardware’ database.

Testing will also be performed on the ‘View Inventory’ function to prevent deadlock. This is to ensure that no two staff members can enter the same description for the same activity at the same time, or that no two entries can be the same with different employee IDs.

Further testing will be required to ensure that there is no information loss in regards to the Customer Satisfaction Survey.

### 2.3 User Procedures

All use cases described in documentation are to be reproduced step-by-step as described by the document by QA personnel. QA personnel are to document and report inconsistencies in documentation during testing. A final QA report will be delivered immediately upon completion to developers for review.

### 2.4 Operator Procedures

Functionality testing, code review, and unit testing are the procedures used to make sure the application can be run and supported in a production environment.

Help Desk procedures are: priority, create tickets, documentation, problem solving, communication, and problem update. In priority the personnel determine what is or what is not a major problem. Then the personnel will create tickets to keep track of an error/problem and document the information. The personnel will attempt to solve the problem and then communicate with the client about the ongoing progress. Lastly the problem will receive evaluation to check if it requires more effort or has been solved.

## 3. Features To Be Tested

|  |  |  |
| --- | --- | --- |
| **Feature(s)** | | **Specification** |
| Backup | Create | Product must send spreadsheet to offsite location for backup. |
|  | Restore | Product must retrieve saved backup from offsite location and implement it in place of current database. |
| Employee | Update | Product must modify entry in database containing Employee data. |
|  | Add | Product must create a new entry in the database containing Employee data. |
|  | Delete | Product must be able to remove entry from database containing Employee data. |
| Supervisor | Update | Product must modify entry in database containing Supervisor data. |
|  | Add | Product must create a new entry in the database containing Supervisor data. |
|  | Delete | Product must be able to remove entry from database containing Supervisor data. |
| Department | Add | Product must create a new entry in the database containing Department data. |
|  | Delete | Product must be able to remove entry from database containing Department data. |
| Inventory | Update | Product must modify entry in database containing Inventory data. |
|  | View | Product must allow a user to view entries in the database containing Inventory data. |
|  | Add | Product must create a new entry in the database containing Inventory data. |
|  | Delete | Product must be able to remove entry from database containing Inventory data. |
| Ratings | Add | Product must create a new entry in the database containing Ratings data. |
|  | Delete | Product must be able to remove entry from database containing Ratings data. |
| Transaction | View | Product must allow a user to view entries in the database containing Transaction data. |
|  | Delete | Product must be able to remove entry from database containing Transaction data. |

## 4. Features Not To Be Tested

|  |  |  |
| --- | --- | --- |
| **Item being tested (System)** | **Feature not being tested** | **Reason(s)** |
| Rating | Update | Once a customer submits a survey it should be final |
| Transaction | Create/Add | New transactions are handled by POS devices that add new lines to the database. POS devices do not have any interaction with our software. |
| Inventory | Low Stock | Would require a notifications page when stock for an item is low |
| Department | Update | Items associated with the Department will require a new form of Database |
| GUI | Aesthetics | The GUI main function is to navigate and display all major areas of the project |
| Excel Database | View | The user of the application will not have access to the database, therefore a view button is not needed |
| Login | New username | An admin or manager should only be able to create an employee credentials once |
| Create Account | Security | The project will not encrypt the username or password stored in a Excel database |

## 5. Approach

### 5.1 Component Testing

Testing is conducted to ensure that the program follows the state-transition diagram(s) as closely as possible. Also we are testing to ensure that all use cases are satisfied and the program’s writing is robust and logically sound.

### 5.2 Integration Testing

We will test the application hardware comparable to that of the user’s. This is to ensure proper functionality in the workplace and to prevent errors during software delivery and maintenance. The systems will be integrated via CD-ROM when the product is delivered to the customer. All testing will be done remotely.

### 5.3 Conversion Testing

Testing is conducted over a secure medium such as GroupMe to ensure that, in the transfer of files between systems, all customer data is secure. Any and all files to be shared in GroupMe or in group meetings are not to be shared or distributed.

### 5.4 Job Stream Testing

The file is tested both as an executable and as a solution file to ensure that the application operates in the production environment. This is an ideal situation for both the developer and the customer, who will interact with the program in different ways.

### 5.5 Interface Testing

The application operates on a laptop as well as on the customer’s machine, or its equivalent.

### 5.6 Security Testing

Running the python code on Pycharm IDE and linking it to excel database and Tkinter (name of GUI) checking each function case and making sure the GUI responds correctly and linking the correct database without loss of data.

### 5.7 Recovery Testing

Creating an Excel database that updates and stores all the current information before deletion. Only the admin will have access to the database, and for demonstration purposes a user will delete everything and let the admin restore everything again.

### 5.8 Performance Testing

Rewriting code to the most proficient way and linking a database that multiple devices (desktops) can use. Since the software will be developed and then stored in a CD Rom it will be very portable.

### 5.9 Regression Testing

We maintain several working versions of our application as backups in case anything goes wrong with the version currently being tested. All working versions are completed before any further changes are made.

### 5.10 Acceptance Testing

The product was tested by the customer on October 30, 2020. Stan’s Hardware has expanded to include 2 new departments: Flooring and Lumber, which will be included in Version 4.

### 5.11 Beta Testing

Multiple versions of the application were tested. The customer’s primary concern was the inclusion of their two new departments in the inventory section of the app. We assured them that the expansion would be implemented in the new version of the application. The pre-release version of the product is a “go”; further testing and implementation of new data is required to bring the newer version(s) up to speed.

## 6. Pass / Fail Criteria

### 6.1 Suspension Criteria

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Activities Suspended** | **Explanation** |
| Thanksgiving (Holiday) | All | Employees are given the Thanksgiving holiday off |
| SH Closure | All | The permanent closure of Stan’s Hardware will trigger a cessation of development. |
| Data Loss | Testing | The loss of software related to the development of the product will result in a suspension of testing until data is recovered. |
| Recurring failure | Testing | An uncontrollable, recurring failure will cease testing until corrected by developers. |

### 6.2 Resumption Criteria

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Activities Resumed** | **Explanation** |
| End of Thanksgiving | All | The day after Thanksgiving is a work day. |
| Data Recovered | Testing | Upon recovery of data, testing may resume. |
| Failure Correction | Testing | Patched failure correction will allow testing to resume. |

### 6.3 Approval Criteria

Test results will be approved on a case-by-case basis in which the case results match the description of expectations for that result.

## 7. Testing Process

One of the methods used in performing test activities is Functional testing. Within this form of testing we used unit, integration, system, and acceptance testing. In unit testing we the developers check the code and make sure each function works correctly with the Tkinter GUI. In integration testing we break down the code into different parts and check only one section at a time. In system testing we check that all parts work together and the software works as intended. In acceptance testing the team leader (Edward) determines if the software is ready for delivery/submission.

### 7.1 Test Deliverables

-Test Strategy: It is a document that explains the approach on how we go about testing the software and achieve the end goal.

-Test Plan: Has The plans for all the testing activities that will be conducted to create a good software.

-Test Data: The data that is used by the devs/testers to run the test cases.

Test Input: What the tester enters to proceed with the testing.

Test Output: What the software must display in console or GUI.

-Bug Report: The purpose of this document is to show steps on how to replicate bugs easily so that devs can know how to get rid of bugs or add a patch.

-Test Summary Report: Report that contains summary of all test activities conducted and test results (final).

-Test Incident Report: all the incidents that are resolved or unresolved during testing.

-Release Notes: Notes sent to client containing a list of new releases and bug fixes

-User Guide: A guide that gives insight to the client on how to access the software application

### 7.2 Testing Tasks

Tasks necessary to prepare are Test Strategy and Test Plan documents. The devs will need to have Pycharm open to test use cases and interact with GUI. Inter Task Dependencies: Finish-Start (Task B cannot start until task A has finished).

### 7.3 Responsibilities

The two developers (Edward and Christopher) manage, design, and prepare, while the two testers (Jordan and Michael) all are responsible for executing, witnessing, checking, and resolving test activities.

### 7.4 Resources

**Testing Tasks**

|  |  |
| --- | --- |
| **Resource** | **Responsibility** |
| Christopher Pando | Creating Test Plan |
| Edward Aranda | Creating Test Strategy |

**Testing Activities**

|  |  |
| --- | --- |
| **Resource** | **Responsibility** |
| Jordan Adriano | Unit Testing |
| Michael Giraldi | Integration Testing |
| Christopher Pando | System Testing |
| Edward Aranda | Acceptance Testing |

### 7.5 Schedule

|  |  |  |
| --- | --- | --- |
| **Task** | **Deliverable** | **Duration** |
| Develop test cases | Requirements Document | October 15 - October 20 |
| Functional testing design | Test Plan Document | October 20 - October 21 |
| Prepare testing requirements | Test Strategy Document | October 21 - October 22 |
| Execute tests | Test Summary Report | October 23 - October 28 |
| Report defects/bugs | Bug Report | October 23 - October 28 |
| Complete test report | Test Plan Document | October 28 - October 30 |

## 8. Environmental Requirements

### 8.1 Hardware

The test environment requires the following hardware:

* 4 Windows 10 PC’s with wifi and capable of running Python scripts

### 8.2 Software

* 4x copies of Windows 10
* 2x installations of Python 3.7
* 2x installations of Pycharm
* 2x installations of Jupyter Notebook
* 4x copies of Microsoft Office: Word, Excel

### 8.3 Security

The test environment does not require any specific security features.

### 8.4 Tools

No specialized tools are necessary for proper testing of our software. All testing can be performed manually using existing scripts.

### 8.5 Publications

* *Object-Oriented and Classical Software Engineering 8th ed*, S. R. Schach
* <https://docs.python.org/3/library/tkinter.html>, tkinter documentation

### 8.6 Risks and Assumptions

Due to the short timeframe left to complete the project and limited meet times, there are significant time constraints to test the product. A significant risk stems from availability of personnel for testing. This will be addressed by dividing-and-conquering testing tasks, such that if a person is unavailable and their work incomplete that tasks will be small enough to be picked up and shared amongst remaining members. The test team makes the assumption that testing hardware is sufficiently similar to the target hardware such that local testing on target hardware will result in minimal performance difference.

## 9. Change Management Procedures

Change is initiated through a formal request for change, submitted to team lead (Edward), for approval to review. Reviews are done through impact analysis: resources required, availability of data, and performance analysis on existing code. Finally, the change is approved or denied by Edward, based on the result of the review.

## 10. Plan Approvals

|  |  |  |
| --- | --- | --- |
| **Name** | **Signature** | **Date of Approval** |
| Edward Aranda |  | 10/14/2020 |
| Edward Aranda |  | 10/20/2020 |
| Michael Giraldi |  | 10/29/2020 |
| Michael Giraldi |  | 10/31/2020 |