

International Organization for Standardization  
Software Design Document (ISO APP)

2016-03-07

International Organization for Standardization	Version <1.0>
Software Design Document	Date 2016-03-07

### Revision History

Date	Version	Description	Author
07/03/2016	<1.0>	Initial Version of Document	Bulumko Wulana

# Software Design Document

## 1. Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case diagrams, sequence diagrams, collaboration models, object behavior models, and other supporting requirements data.

### 1.1 Purpose

The purpose of this document is to provide the software developer with enough information to be able to understand and be able build the required application based on the detailed information provided in this document.

### 1.2 Scope

The Software Design Document serves as the foundation on which the whole system will be built on. The system being built here will be used in conjunction with the current ICTWorx Intranet system. This application will guide the user through the necessary steps involved in ensuring that a company/organization is ISO accredited, however the application does not accredit the company/organization. It simply guides the user through the process by giving them instructions on what needs to be done and when. The rest of the document will go into detail as to how the application will work.

### 1.3 Definitions, Acronyms, and abbreviations

- **ISO** - International Organization for Standardization is an independent, non-governmental organization, the members of which are the standards organization of the 164 member countries. It is the world's largest developer of voluntary international standards and facilitates world trade by providing common standards between nations. Nearly twenty thousand standards have been set covering everything from manufactured products and technology to food safety, agriculture and healthcare.
- **DB** – refers to the Database object
- **DBMS** - is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases.
- **RDBMS** - A relational database management system (**RDBMS**) is a database management system (**DBMS**) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. In 2015, many of the databases in widespread use are based on the relational database model.

### 1.4 References

- ISO Application Software Design Document
  - Version 1.0, Last updated March 7, 2016

### 1.5 Overview

This document is divided into the following sections.

1. Introduction
2. Glossary
3. Use Cases
4. Design Overview
5. System Object Model
6. Object Description
7. Object Collaboration
8. Data Design
9. Dynamic Model
10. Non-functional requirements
11. Supplementary Documentation

### **3. Use Cases**

#### **Use-Case Model Survey**

##### **3.1 Actors**

3.1.1 Admin – This user is responsible for daily maintenance of the system, running queries on the DB, updating user's profiles, generating reports, backups, report bugs to developers, configuring the application.

3.1.2 Managers – This user is responsible for the following, create tasks, assign tasks, and approve task completion, generating reports about tasks and employees working under them, view processes, update process data, download records of the company.

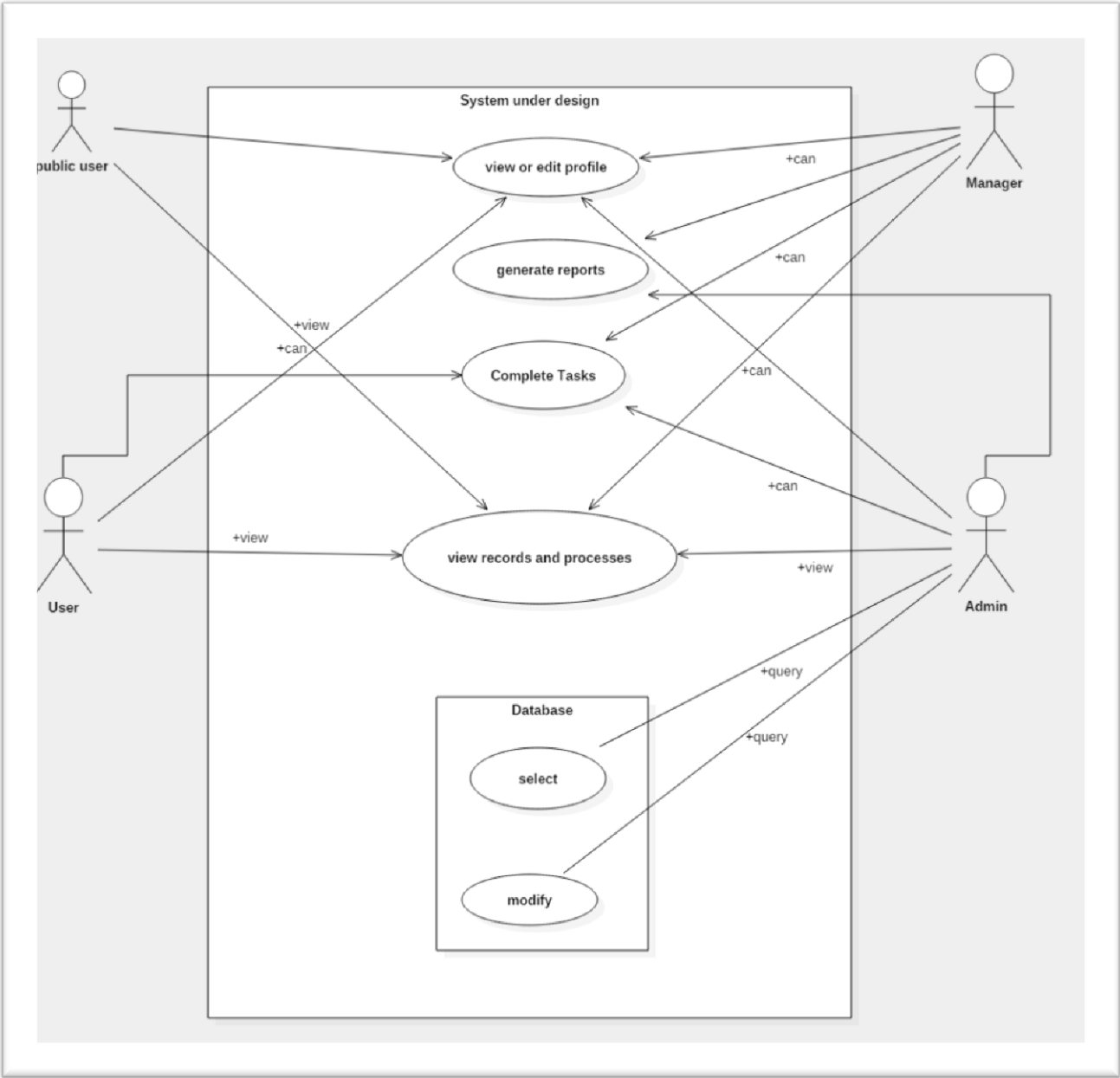
3.1.3 User – This user has minimal access, and is the default user when creating an account. This user is responsible for the following, completing task and uploading documents as proof of completion of task, modify their profile.

3.1.4 System under design – This actor is the system being built and is what will be communicating with most of the actors above.

3.1.5 Public user – This actor represents anyone who accesses the system and does not fall under any of the specified users above.

##### **3.2 Use-Case diagram**

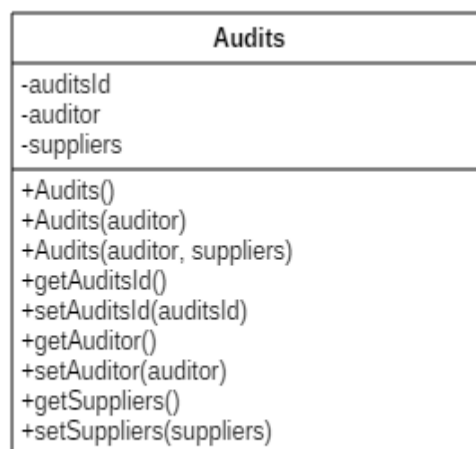
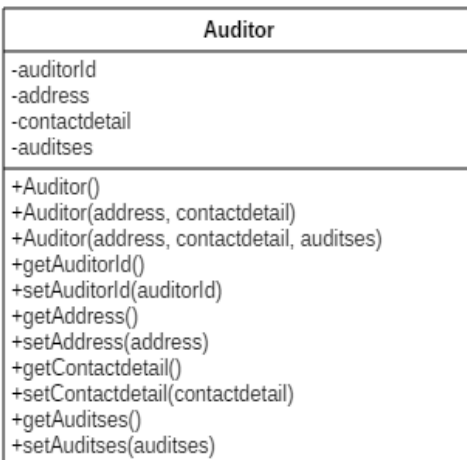
Use Case Diagram for ISO APP

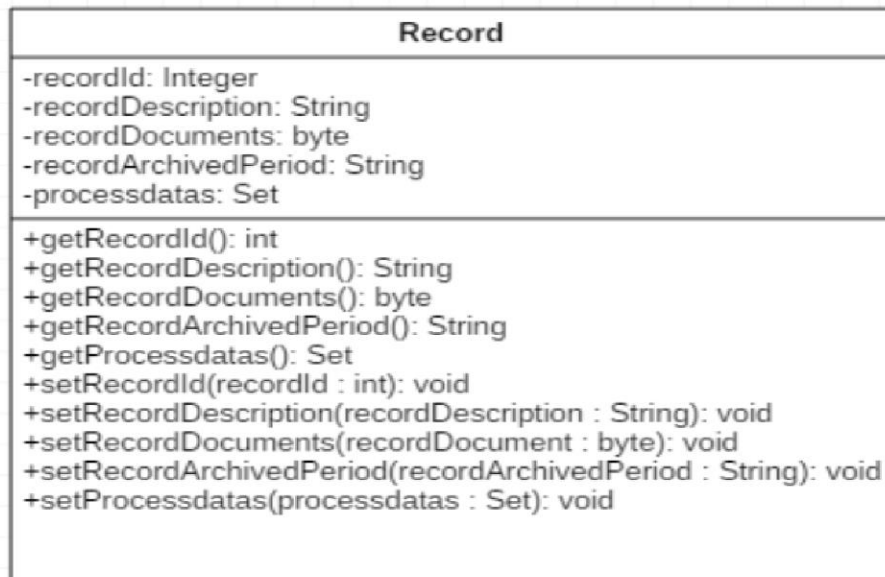
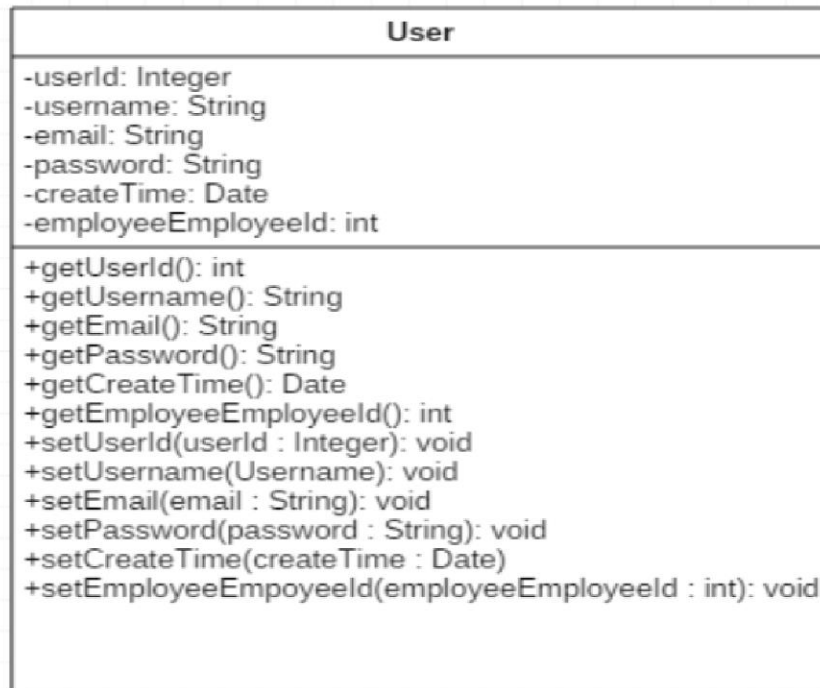


## Use Case Diagram for ISO APP (Admin user)



## 4. UML Class Diagrams







Question
-questionId: Integer -questionData: String -tabTabId: int
+getQuestionId(): int +getQuestionData(): String +getTabTabId(): int +setQuestionId(questionId : int): void +setQuestionData(questionData : String): void +setTabTabId(tabTabId : int): void

ProcessRisk
-processRiskId: Integer -processRiskDescription: String -processRiskType: String -processRiskResponse: String -processRiskDueDate: Date -processRiskClosedDate: Date -processRiskCurrentControls: String -process: Set
+getProcessRiskId(): int +getProcessRiskDescription(): String +getProcessRiskType(): String +getProcessRiskResponse(): String +getProcessRiskDueDate(): Date +getProcessRiskClosedDate(): Date +getProcessRiskCurrentControls(): String +getProcesses(): Set +setProcessRiskId(processRiskId: int): void +setProcessRiskDescription(processRiskDescription: String): void +setProcessRiskType(processRiskType: String): void +setProcessRiskResponse(processRiskResponse: String): void +setProcessRiskDueDate(processRiskDueDate: Date): void +setProcessRiskClosedDate(processRiskClosedDate: Date): void +setProcessRiskCurrentControls(processRiskCurrentControls: String): void +setProcesses(processes: Set): void

ProcessData
-processDataId: int -instruction: Instruction -record: Record -processDataDescription: String -processDataReviewFrequency: String -processDataArchivedPeriod: String -processDataAuth: int -processDataDocument: byte -processes: Set
+getProcessDataId(): int +getInstruction(): Instruction +getRecord(): Record +getProcessDataDescription(): String +getProcessDataReviewFrequency(): String +getProcessDataArchivedPeriod(): String +getProcessDataAuth(): int +getProcessDataDocument(): byte +getProcesses(): Set +setProcessDataId(processDataId: int): void +setInstruction(instruction: Instruction): void +setRecord(record: Record): void +setProcessDataDescription(processDataDescription: String): void +setProcessDataReviewFrequency(processDataReviewFrequency: String): void +setProcessDataArchivePeriod(processDataArchivePeriod : String): void +setProcessDataAuth(processDataAuth : int): void +setProcessDataDocument(processDataDocument : byte): void +setProcesses(processes : Set): void

Answer
-answerId: int -answerDoc: byte -questionQuestionId: int -answerDateLastModified: Date -authorAuthorId: int
+getAnswerId(): int +getAnswerDoc(): byte +getQuestionQuestionId(): int +getAnswerDateLastModified(): Date +getAuthorAuthorId(): int +setAnswerId(answerId: int): void +setAnswerDoc(answerDoc: byte): void +setQuestionQuestionId(questionQuestionId: int): void +setAnswerDateLastModified(answerDateLastModified: Date): void +setAuthorAuthorId(authorAuthorId: int): void

Author
-authorId -address -contactdetail -informationInformationId +instructions
+Author() +Author(address, contactdetail, informationInformationId) +Author(address, contactdetail, informationInformationId, instructions) +getAuthorId() +setAuthorId(authorId) +getAddress() +setAddress(address) +getContactdetail() +setContactdetail(contactdetail) +getInformationInformationId() +setInformationInformationId(informationInformationId) +getInstrusctions() +setInstructions(instructions)

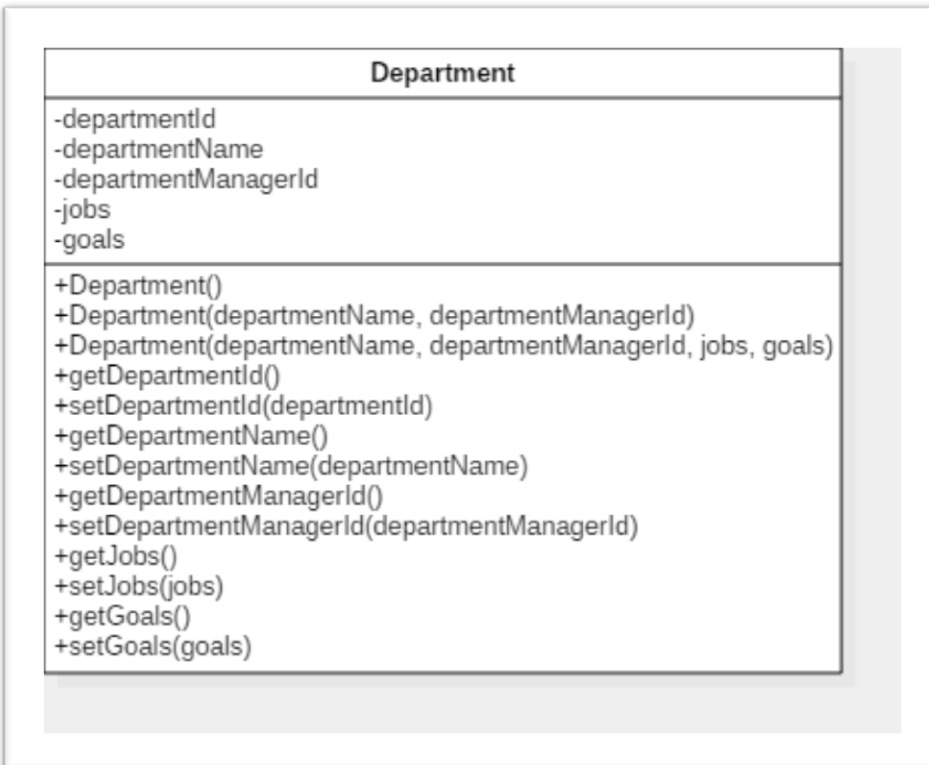
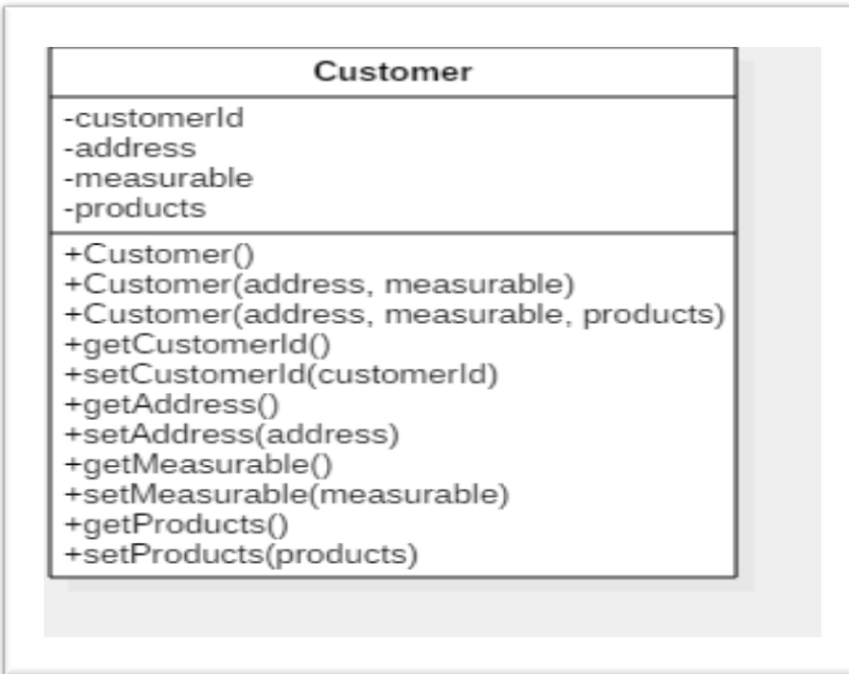
Goal
-goalId -department -process
+Goal() +Goal(department, process) +getGoalId() +setGoalId(goalId) +getDepartment() +setDepartment(department) +getProcess() +setProcess(process)

Tab
-tabId: Integer -tabTitle: String -isoStandardIsoStandardId: int
+getTabId(): int +getTabTitle(): String +getIsoStandardIsoStandardId(): int +setTabId(tabId : Integer): void +setTabTitle(tabTitle : String): void +setIsoStandardIsoStandardId(isoStandardIsoStandardId: int): void

Supplier
-supplierId: Integer -audits: Audits
+getSupplierId(): int +getAudits(): Audits +setSupplierId(supplierId : Integer): void +setAudits(audits : Audits): void

Period
-periodId: int -periodStartDate: Date -periodEndDate: Date -periodCompletionPercent: double
+getPeriodId(): int +getPeriodStartDate(): Date +getPeriodEndDate(): Date +getPeriodCompletionPercent(): double +setPeriodId(periodId: int): void +setPeriodStartDate(periodStartDate: Date): void +setPeriodEndDate(periodEndDate: Date): void +setPeriodCompletionPercent(periodCompletionPercent: double): void

Contactdetail
-contactDetailId -contactDetailCellNumber -contactDetailEmailAddress -contactDetailAlternativeMail -contactDetailAlternativeCellNumber -authors -employees -auditors
+Contactdetail() +Contactdetail(contactDetailCellNumber, contactDetailEmailAddress) +Contactdetail(contactDetailCellNumber, contactDetailEmailAddress, contactDetailAlternativeMail, contactDetailAlternativeCellNumber, authors, employees, auditors) +getContactDetailId() +setContactDetailId(contactDetailId) +getContactDetailCellNumber() +setContactDetailCellNumber(contactDetailCellNumber) +getContactDetailEmailAddress() +setContactDetailEmailAddress(contactDetailEmailAddress) +getContactDetailAlternativeMail() +setContactDetailAlternativeMail(contactDetailAlternativeMail) +getContactDetailAlternativeCellNumber() +setContactDetailAlternativeCellNumber(contactDetailAlternativeCellNumber) +getAuthors() +setAuthors(authors) +getEmployees() +setEmployees(employees) +getAuditors() +setAuditors(auditors)



### Employee

-employeeId  
-address  
-contactdetail  
-job  
-employeeContract  
-employeeDocuments  
-informationInformationId  
-managers

+Employee()  
+Employee(address, contactdetail, job, employeeContract, employeeDocuments, informationInformationId)  
+Employee(address, contactdetail, job, employeeContract, employeeDocuments, informationInformationId, managers)  
+getEmployeeId()  
+setEmployeeId(employeeId)  
+getAddress()  
+setAddress(address)  
+getContactdetail()  
+setContactdetail(contactdetail)  
+getJob()  
+setJob(job)  
+getEmployeeContract()  
+setEmployeeContract(employeeContract)  
+getEmployeeDocuments()  
+setEmployeeDocuments(employeeDocuments)  
+getInformationInformationId()  
+setInformationInformationId(informationInformationId)  
+getManagers()  
+setManagers(managers)

### Information

-informationId  
-informationName  
-informationSurname  
-informationIdNumber

+Information()  
+Information(informationName, informationSurname, informationIdNumber)  
+getInformationId()  
+setInformationId(informationId)  
+getInformationName()  
+setInformationName(informationName)  
+getInformationSurname()  
+setInformationSurname(informationSurname)  
+getInformationIdNumber()  
+setInformationIdNumber(informationIdNumber)

## Instruction

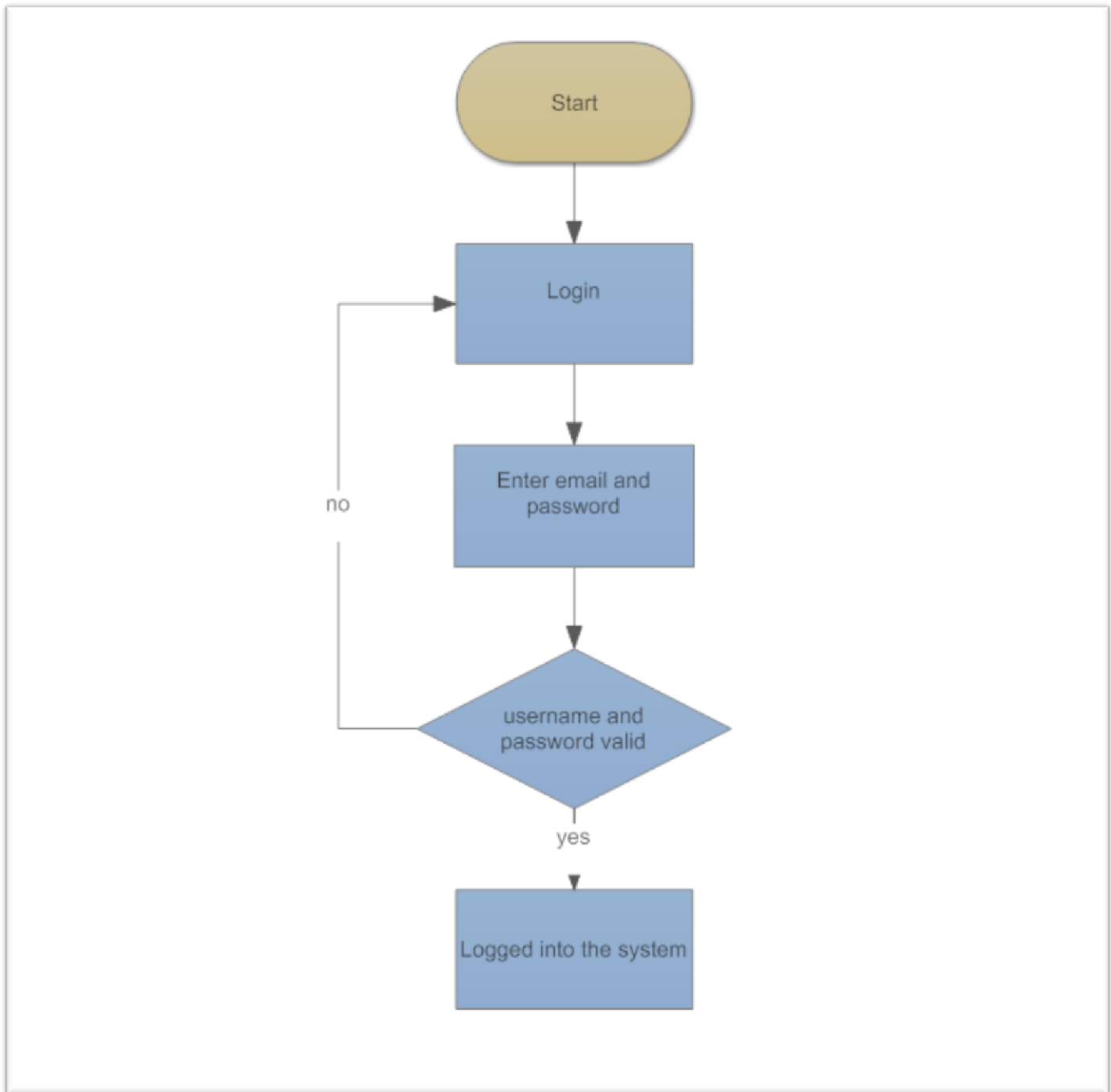
```
-instructionId: Integer
-author: Author
-instructionOldNumber: String
-instructionDescription: String
-instructionArchivedPeroid: String
-processdates: Set

+Instruction()
+Instruction(instructionId : Integer, author : Author)
+Instruction(instructionId, author, instructionOldNumber, instructionDescription, instructionArchivePeroid, processdates)
+getInstructionId()
+setInstructionId(instructionId : Integer)
+getAuthor()
+setAuthor(author : Author)
+getInstructionOldNumber()
+setInstructionOldNumber(instructionOldNumber : String)
+getInstructionDescription()
+setInstructionDescription(instructionDescription: String)
+getInstructionArchivedPeroid()
+setInstructionArchivedPeroid(instructionArchivedPeroid : String)
+getProcessdates()
+setProcessdates(processdates: Set)
```



## 5. Flow Charts

### 5.1 Login Flow Chart



## 5.2 SHEQ Road Map Flow chart

