



Biographical Affidavit

Report #3

Group 1:

Hipolito Bautista

Justin Chuc

Michael Gomez

Rene Allen

Dennis Arnold

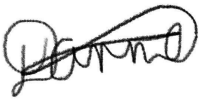



Jahmur Lopez

CMPS4131

Manual Medina

March 31,2023

TEAM SIGNATURES

SIGNATURE BLOCK			
Statement	I did my share of the work, and I have a general understanding of the contents of the assignment.		
Team Member	Contribution	Signature	Date
Dennis Arnold	<ul style="list-style-type: none"> • Use cases • Traceability matrix • System operation contract • OCL contract specification 		17/5/2023
Jahmur Lopez	<ul style="list-style-type: none"> • Design of test cases • Interaction diagram description • Template of report 3 		17/5/2023
Justin Chuc	<ul style="list-style-type: none"> • 	<i>J. Chuc</i>	17/5/2023
Hipolito Bautista	<ul style="list-style-type: none"> • Use cases • Interaction Diagrams • Requirement specification • Package diagram • Package diagram description 		17/5/2023
Michael Gomez	<ul style="list-style-type: none"> • Team Leader, Logistics • Problem Statement • Document Management • History of Work • Attribute Definitions • Architectural Styles and Data structures 		17/5/2023
Rene Allen	<ul style="list-style-type: none"> • Conceptual Model • Concept Definitions • Class Diagram and Interface Specifications 	<i>Rene Allen</i>	17/5/2023

INDIVIDUAL CONTRIBUTION BREAKDOWN

RESPONSIBILITY LEVEL		Team Members						Point Total
		Dennis Arnold	Jahmur Lopez	Justin Chuc	Hipolito Bautista	Michael Gomez	Rene Allen	
	Section 1: Customer Statement (6 points)					100%		6
	Section 2: Glossary (4 points)		100%					4
	Section 3: System Requirements (6 points)	48%	1%	1%	48%	1%	1%	6
	Section 4: Functional Requirements Specification (30 points)	30%	10%		35%		25%	30
	Section 5: Effort Estimation (4 points)	100%						4
	Section 6: Domain Analysis (25 points)	40%				20%	40%	25
	Section 7: Interaction Diagrams (40 points)	20%			40%		40%	40
	Section 8: Class Diagrams & Interface Specification (20 points)	20%			40%		40%	20
RESPONSE	Section 9: System Architecture and Design (15 points)	20%	20%		20%	20%	20%	15
	Section 10: Algorithms and Data Structure (4 points)					100%		4

S I B I L I T Y L E V E L	Section 11: User Interface Design and Implementation (11 points)	50%	50%					11
	Section 12: Test Case Design (12 points)		100%					12
	Section 13: History of Work (5 points)					100		5
	Section 14: Project Management (13 points)					100		13
	Section 14: References (5 points)	100%						5
	Total:							200

TABLE OF CONTENTS

TEAM SIGNATURES	2
INDIVIDUAL CONTRIBUTION BREAKDOWN	3
TABLE OF CONTENTS	5
SUMMARY OF CHANGES	7
CUSTOMER STATEMENT OF REQUIREMENTS	8
Problem Statement:	8
Glossary of Terms:	11
SYSTEM REQUIREMENTS	12
Functional Requirements:	12
Non-Functional Requirements:	12
On-Screen Appearance Requirements:	13
FURPS Table:	17
FUNCTIONAL REQUIREMENTS	18
Stakeholders:	18
Actors and Goals:	18
Use Cases	19
Casual Description	19
Use Case Diagram	21
Traceability Matrix	22
Fully-Dressed Use Case Description	23
System Sequence Diagrams:	31
EFFORT ESTIMATION	41
DOMAIN ANALYSIS	45
Concept Definition	49
Association Definition:	52
Attribute Definition:	52
Traceability Matrix:	53
System Operation Contract:	54
Mathematical Model / Algorithms:	57
INTERACTION DIAGRAMS	58
CLASS DIAGRAMS AND INTERFACE SPECIFICATIONS	67
Design Patterns:	69
OCL Contract Specification:	69
SYSTEM ARCHITECTURE AND SYSTEM DESIGN	70
Identifying Subsystems & Architecture Styles:	70
UML Package Diagram	70
Mapping Subsystems to Hardware:	71
Connectors and Network Protocols:	71

Global Control Flow:	72
Hardware Requirements:	72
ALGORITHMS AND DATA STRUCTURES	73
USER INTERFACE DESIGN AND IMPLEMENTATION	74
Modifications:	74
GUI Designs:	75
Ease of Use:	81
DESIGN OF TESTS	82
HISTORY OF WORK	90
REFERENCES	91

SUMMARY OF CHANGES

1. Formatted document to be cohesive
2. Updated Use Cases and their derived components
3. Updated Requirement numerations
4. Updated Traceability Matrices
5. Updated Sequence Diagrams
6. Updated Use Case Diagrams

CUSTOMER STATEMENT OF REQUIREMENTS

Problem Statement:

OSIPP(the Office of the Supervisor of Insurance and Private Pension), a part of the Ministry of Finance, which is a part of the Government of Belize. As a regulatory body, it's required to have information for the purpose of due diligence and ensuring insurance companies and pension administrators properly follow the regulations.



OFFICE OF THE SUPERVISOR OF
INSURANCE & PRIVATE PENSION

The primary purpose of the Biographical Affidavit is to gather information on the various people involved with entities that OSIPP is tasked with regulating. These people are usually:

- Shareholders
- Directors
- Senior Management
- Compliance Officers
- Insurance Agents
- Insurance Brokers
- Pension Administrators

The biographical affidavit is thorough, with the information needed for its completion. A form may be sent back multiple times because the information is often incomplete or incorrect. Leading to the submission of the biographical affidavit itself being time-consuming and bothersome since both parties would prefer to perform other tasks.



In addition to submitting the biographical affidavit taking quite a bit of time, the information is extracted for future reference and proper information management.

Since the form is currently being physically submitted, handwritten and typed submissions are received. Handwritten forms are problematic because often, the writing is illegible. While typed forms still need to be manually extracted. Both situations lead to being wasted on what should be a minor task at OSIPP.

OSIPP has already digitized some of its processes in the past; The electronic financial reports and AML(Anti-Money Laundering) risk reports are some of these two. The digitization of these forms has improved efficiency at OSIPP since these are done more frequently than the other forms. Thus saving space for the remaining forms that they process. It's now time for the biographical affidavits to be digitized; it's long overdue since the first attempt was back in 2018.



OSIPP is eagerly awaiting the improvements this project will bring to process the biographical affidavit. The digitization of this form will significantly increase the efficiency of the process since no physical paper processing will need to be done. Since no paper is involved throughout the process, it's less bothersome for the staff who will have to manage the files. Having the information available quickly will significantly improve the speed of the reviews when performed; no more digging through boxes of papers to find the correction file.

We'd like the digital form to mimic the other digitized forms, where instead of scanning a document and sending it via email, the people who need to fill out the forms will be able to do so with a digital form over a traditional one. We'd also like to add the ability to add comments to note any changes made to the information; for internal use, of course. An example would be; a member of staff reviewing a previously processed biographical affidavit that needs to be updated. They'll note any changes that have been made so that, down the line, future staff will be able to observe what has been changed.

A current issue with the other digitized forms is that sometimes the information they contain must be made physical. We'd like to have the ability to print both a tabulated and pdf

version of the biographical affidavit information. They'll be seldom used, but when the need arises, it's usually because the information is currently needed elsewhere. Composing files with that information manually is time-consuming; therefore, being able to print like that is important to us.

Glossary of Terms:

Terms	Meaning
Shareholders	A person or institution that has invested money in a corporation in exchange for a “share” of the ownership
Directors	Someone elected or appointed to manage a company’s business and affairs
Senior Management	Generally individuals at the highest level of management of an organization who have the day-to-day task of managing that organization; sometimes a company or a corporation.
Compliance Officers	are responsible for ensuring that all corporate processes and procedures comply with the law
Insurance Agents	a person who solicits, negotiates, or instigates insurance contracts on behalf of an insurer and can be independent or an employee of that insurer.
Insurance Brokers	An intermediary who sells, solicits, or negotiates insurance on behalf of a client for compensation.
Pension Administration	The act of performing various types of yearly service on an organizational retirement plan.
Internal use	A system used within an organization whose data or processes are not intended for the public.
External use	A system used by actors outside the organization
Digitization	The process of converting information to a digital format
OSIPP	Acronym for “The Office of the Supervisor of Insurance & Private Pension”

SYSTEM REQUIREMENTS

Functional Requirements:

Identifier	PW	Requirement
REQ-1	5	The system shall allow the internal user to login to access the Digital Biographical Affidavit.
REQ-2	5	The system shall allow the external user to fill in the digitized form with the necessary information.
REQ-3	5	The system shall allow the external user to send the form electronically after all information is filled out.
REQ-4	4	The system shall allow the internal user to verify if the form was filled out correctly.
REQ-5	3	The system will archive outdated forms to preserve the records.
REQ-6	5	The system shall allow OSIPP public officers to search for Affidavit data.

Non-Functional Requirements:

Identifier	PW	Requirement
NONREQ-1	2	The system shall provide a non-cluttered, user-friendly, easy-to-understand web application.
NONREQ-2	4	The system shall use an algorithm to transfer user's information to the database and back to the user.
NONREQ-3	5	The system shall provide a unique code to their work email to access the web application.
NONREQ-4	3	The system shall have a holding database to store a form for verification.
NONREQ-5	5	The system shall allow the administrator to verify the forms before having them stored in the database.
NONREQ-6	5	The system shall generate a unique code to associate each form created.
NONREQ-7	2	The system shall store information from the electronic forms on a

		database.
NONREQ-8	5	The system shall allow the user to be warned about incorrect information on the form to be fixed or the field being left blank.

On-Screen Appearance Requirements:

Identifier	PW	Requirement
ONSREQ-1	5	The design of the system shall be related to the color scheme of OSIPP.
ONSREQ-2	4	The system shall display a unique code on the top right of the form.



OFFICE OF THE SUPERVISOR OF
INSURANCE & PRIVATE PENSION

Welcome

Username

Enter your username

Password

Enter your password

Log In

Cancel

[Forgot your password?](#)

Figure 1. Showing the on-screen requirements: ONSREQ-1



BIOGRAPHICAL AFFIDAVIT FORM

1. Affiant's Full Name:
2. Other names used at any time:
3. Have you ever had your name changed? ☐ Yes ☒ No
If "Yes", (a) provide previous name(s) and (b) give the reason for the change:
4. Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.

Document	Number of Document	Date of issuance	Country of Issuance
Social Security	23262988	2/15/21	Belize
Passport	P00035684	10/28/20	Belize
National Health Insurance	N/A	N/A	N/A
Other, specified			

5. Date of Birth: (DD/MM/YYYY)
6. Place of Birth: include District/State and Country)
7. Nationality: Indicate how acquired.
☒ Birth
☐ Naturalization
☐ Marriage
☐ Other, specify

8. Spouse's Name:
9. Affiant's Address:

Physical Address	Telephone No.	Fax No.	Email Address
Residential	501-223-4567	N/A	johndoe@gmail.com
Business	N/A	N/A	N/A

Signature:

Date:

SUBMIT

Figure 2. Showing the on-screen requirements: ONSREQ-1, and ONSREQ-2



Form Review & Approval

ID: 2045783820

Comment

- Affiant's Full Name:
- Other names used at any time
- Have you ever had your name changed? ☐ Yes ☒ No
If "Yes", (a) provide previous name(s) and (b) give the reason for the change:
- Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.

Document	Number of Document	Date of issuance	Country of issuance
Social Security	23262998	2/15/21	Belize
Passport	P00035684	10/28/20	Belize
National Health Insurance	N/A	N/A	N/A
Other, specified			

- Date of Birth: (DD/MM/YYYY)
- Place of Birth: include District/State and Country)
- Nationality: Indicate how acquired.

☒ Birth

☐ Naturalization

☐ Marriage

☐ Other, specify

- Spouse's Name:
- Affiant's Address:

Physical Address	Telephone No.	Fax No.	Email Address
Residential	501-223-4567	N/A	john.doe@gmail.com
Business	N/A	N/A	N/A

Signature:

Date:

Approve

Decline

Figure 3. Showing the on-screen requirements: ONSREQ-1

FURPS Table:

Functionality	<ul style="list-style-type: none">● Users will be able to include their signature at the end of the form.
Usability	<ul style="list-style-type: none">● The web application will have a login screen to access the form.● Input boxes are clearly labeled and formatted
Reliability	<ul style="list-style-type: none">● Checks are done to verify if the information has been entered correctly or if there's missing information● Version control of the application will be done for upkeep of the
Performance	<ul style="list-style-type: none">● Multiple users can use the application simultaneously
Supportability	<ul style="list-style-type: none">● Website Application runs on many supported browsers.

FUNCTIONAL REQUIREMENTS

Stakeholders:

A stakeholder is an individual or group with an interest in the operations of an organization. For improved file administration within the sub-department of the Ministry of Finance, the digitized version of the Biographical Affidavit has been created for the Office of the Supervisor of Insurance & Private Pension. Consequently, the following individuals are involved in this process:

- 1.) A site administrator that shall manage all system's data (database and front-end information).
- 2.) The agent shall interact with the system to complete the form.
- 3.) Public Officers of OSIPP shall verify the form that was filled out, to ensure all information is in order

Actors and Goals:

- **Site Administrator**- can change the entire system and manipulate every piece of data. These comprise database details, the front-end interface, and any system logs produced.
- **3rd Party Agents** - can access the system to fill out the biographical affidavit form on the system so it can be sent for verification before having it stored in the database.
- **Public Officers of OSIPP**- can access customer's form to verify if all fields were entered correctly and none were left blank; they'll also be able to access the information stored in the database for their own purposes.

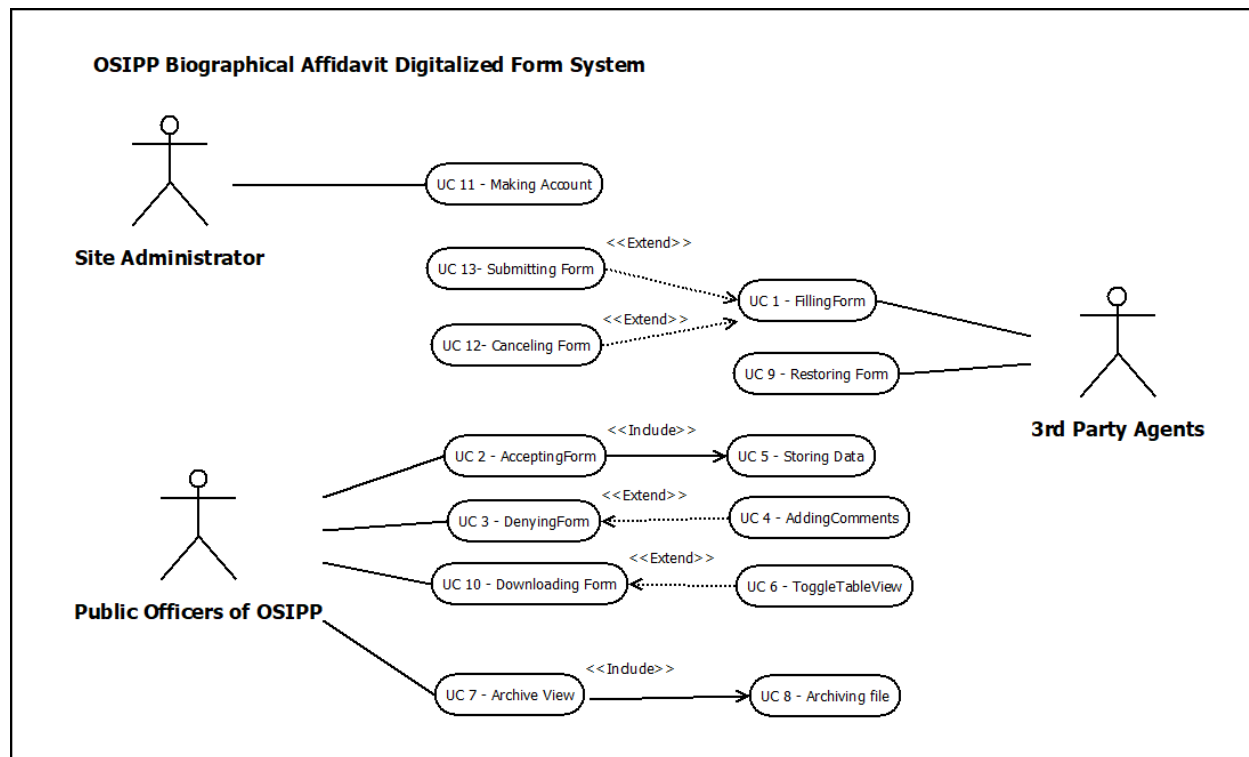
Use Cases

Casual Description

Name	Description	Requirement Covered
UC-1: FillingForm	Form being filled out	REQ-2, ONSREQ-2 ,NONREQ-8
UC-2: AcceptingForm	Accept a submitted form	REQ-3, REQ-4, NONREQ-5
UC-3: DenyingForm	Denying a submitted form	REQ-3, REQ-4, NONREQ-5, NONREQ-6
UC-4: AddingComments	Add comments to a submitted form	REQ-1, REQ-4, NONREQ-5, ONSREQ-1
UC-5: StoringData	Storing the information to the database	REQ-1,REQ-4
UC-6: TogglingTableView	View all accepted form in a readable manner	REQ-1, NONREQ-1

UC-7: ViewArchive	View all archived forms	REQ-2,REQ-3
UC-8: ArchivingFile	Archive a form	REQ-2,REQ-3
UC-9: RestoringForm	Restore a form that has been submitted	REQ-1, NONREQ-1
UC-10: DownloadForm	Download a form	REQ-1, NONREQ-1
UC-11: MakingAccount	Create account for public officers	REQ-1, NONREQ-3
UC-12: CancelingForm	Close a form	REQ-2, NONREQ-8
UC-:13 Submitting form	Submit a form	REQ-1,REQ-2 ,REQ-3

Use Case Diagram



Traceability Matrix

PW Value		UC 1 - FillingForm	UC 2 - AcceptingForm	UC 3 - DenyingForm	UC 4 - AddingComments	UC 5 - StoringData	UC 6 - TogglingTableView	UC 7 - View Archive	UC 8 - ArchivingFile	UC 9 - RestoringForm	UC 10 - DownloadForm	UC 11- MakingAccount	UC 12 - CancelingForm	UC 13 - SubmittingForm
5	REQ-1				X	X	X			X	X	X		X
5	REQ-2	X						X	X				X	X
5	REQ-3		X	X				X	X					X
4	REQ-4		X	X	X									
3	REQ-5					X								
5	REQ-6													
2	NONREQ-1						X			X	X			
4	NONREQ-2													
5	NONREQ-3											X		
3	NONREQ-4													
5	NONREQ-5		X	X	X									
5	NONREQ-6			X										
2	NONREQ-7													
5	NONREQ-8	X											X	
5	ONSREQ-1				X									
4	ONSREQ-2	X												
	Total Weight	14	14	19	19	8	7	10	10	7	7	10	10	15

Fully-Dressed Use Case Description

The use cases that received the highest priority weight in the traceability matrix underwent the fully-dressed description. UC-2, UC-3, UC-4 ,UC-5, UC-6, UC-7, UC-8 ,UC-9, UC-10, UC-11, UC-13 and UC-14 were considered because other use cases such as UC-1 include the processes carried out in these use cases.

Use Case UC-1	FillingForm
Related Requirements	REQ-2, NONREQ-8, ONSREQ-2
Initiating Actor	3rd Party Agents
Actor's Goal	Fill out form with personal information
Participating Actors	3rd Party Agents
Preconditions	Open Form
Postconditions	Submit form after properly filling all fields
Flow of Events for Main Success Scenario: →1 3rd party agent opens the form. →2 3rd party agent fills in the form with their personal information. →3 3rd party agent submits form .	

Use Case UC-2	AcceptingForm
Related Requirements	REQ-3, REQ-4, NONREQ-5
Initiating Actor	Public Officers of OSIPP
Actor's Goal	Confirm the document is properly filled out.
Participating Actors	3rd Party Agents
Preconditions	Form has been submitted for verification by a

	3rd party user. Public Officers of OSIPP are logged in.
Postconditions	Form will be transferred from “unverified forms” to the “verified forms” table in the database.
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP opens the form that was submitted by a 3rd-party agent for revision. →2. Public officer of OSIPP Clicks on the “Accept” button available to him. ←3. System will move form from “unverified forms” to “verified forms” in the database. ←4. System will display a “success” message to the OSIPP Officer	

Use Case UC-3	DenyingForm
Related Requirements	REQ-3, REQ-4, NONREQ-5, NONREQ-6, ONSREQ-1
Initiating Actor	Public Officers of OSIPP
Actor’s Goal	Informs the 3rd party agent that the form has been denied via email with a restoration code. Officers of OSIPP are logged in.
Participating Actors	3rd party agents
Preconditions	Form has been submitted for verification by a 3rd party user. Public Officers of OSIPP are logged in.
Postconditions	
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP opens the form that was submitted by a 3rd-party agent for revision.	

- 2. **Public officer of OSIPP** include::AddingComments(UC4)
- 3. **Public officer of OSIPP** clicks on “deny” button.
- ←4. **System** will display a “form has been denied” message to the OSIPP Officer
- ←5. **System** Sends email to the 3rd party agent that the form has been denied. Within this email a restoration code will be included along with steps on how to utilize it.

Use Case UC -4	AddingComments
Related Requirements	REQ-1, REQ-4, NONREQ-5, ONSREQ-1
Initiating Actor	Public Officers of OSIPP
Actor’s Goal	Indicate fields which have been improperly filled out. Officers of OSIPP are logged in.
Participating Actors	None
Preconditions	Form has been submitted for verification by a 3rd party user. Public Officers of OSIPP are logged in.
Postconditions	Form will be commented by Public officers of OSIPP
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP opens the form in order to use the “add comment” button to place remarks beside the chosen field(s). ←2. System will add a comment field beside the field for the OSIPP officer to input into →3. Public officer of OSIPP will add comments ←4. System will save data inputted in the new comments field by the OSIPP officer.	

Use Case UC -5	StoringData
Related Requirements	REQ-1,REQ-4
Initiating Actor	Public Officers of OSIPP
Actor's Goal	To submit form to be saved in the database
Participating Actors	None
Preconditions	Form fields have been entered correctly and verification was successful.
Postconditions	Form information will be in the database for viewing.
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP Clicks on the “Approve” button available to him. ←2. System will display a “success” message to the OSIPP Officer ←3. System will send record to the stored in database →4. Public officer of OSIPP can view stored records in database	

Use Case UC -6	TogglingTableView
Related Requirements	REQ-1, REQ-6
Initiating Actor	Public Officers of OSIPP
Actor's Goal	Displays approved affidavits
Participating Actors	None
Preconditions	The user is logged into their OSIPP officer account
Postconditions	The affidavits the user has searched for will be listed.
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP searches for an affidavit	

- ←2. **System** will search through the data in the “approved” affidavits table.
- ←3. **System** will display all affidavits which match the users search.

Use Case UC -7	ViewArchive
Related Requirements	REQ-1, REQ-6
Initiating Actor	Public Officers of OSIPP
Actor’s Goal	View all archived affidavits.
Participating Actors	None
Preconditions	OSIPP officer is logged in.
Postconditions	Show all archived affidavits
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP selects “View Archived Affidavits” in the menu. ←2. System Loads user into new page ←3. System Loads all affidavits saved in the “Archived affidavits” table from the database.	

Use Case UC -8	ArchivingFile
Related Requirements	REQ-1, REQ-6
Initiating Actor	Public Officers of OSIPP
Actor’s Goal	Archive affidavit currently being accessed.
Participating Actors	None
Preconditions	View list of files that were approved.
Postconditions	Remove file from the “approved” table and place it in the “archived affidavits” table.
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP selects “View Archived Affidavits” in the menu (UC-7).	

- 2. **Public officer of OSIPP** presses “archive” button next to the affidavit they wish to archive
- ←3. **System** Moves the affidavit from the “approved” table into the “archived affidavits” table
- ←4. **System** Deletes the affidavit from the “approved” table.
- ←5. **System** Informs the OSIPP office that the affidavit was successfully archived.

Use Case UC-9	RestoringForm
Related Requirements	REQ-2,NONREQ-6, NONREQ-8
Initiating Actor	3rd party agents
Actor’s Goal	Restore the form using the restoration code they received via email.
Participating Actors	None
Preconditions	Form must be submitted for review
Postconditions	Previously rejected form is loaded with it’s comments.
Flow of Events for Main Success Scenario: →1. 3rd party agent presses the “Restore Previous Form” button. ←2. System Prompts user for Form ID via popup with instructions. →3. 3rd party agent enters the ID /registration code. ←4. System Loads the form associated with the Form ID with its comments.	

Use Case UC-10	DownloadForm
Related Requirements	REQ-2,NONREQ-6, NONREQ-8
Initiating Actor	Public Officers of OSIPP

Actor's Goal	To download the affidavit as a PDF
Participating Actors	None
Preconditions	OSIPP Officer is in PDF view
Postconditions	Affidavit has been downloaded as a PDF
Flow of Events for Main Success Scenario: →1. Public officer of OSIPP Presses “Download” button available beside all affidavits. ←2. System Gets all data associated with affidavit and converts to PDF. ←3. System Initiates download to OSIPP officer's local machine.	

Use Case UC-11	Making Account
Related Requirements	REQ-1
Initiating Actor	Site Administrator
Actor's Goal	Register Accounts for OSIPP Officers
Participating Actors	Public officer of OSIPP
Preconditions	Must have the work issued email address of OSIPP officer
Postconditions	Registered OSIPP officer account
Flow of Events for Main Success Scenario: →1. Site Administrator clicks “Create Account” to add the OSIPP officer's email address into the “email address” section of the “users” table. →2. Site Administrator adds OSIPP officer's password into the “password” section of the “users” table. ←3 System stores accounts created by the site administrator on the database.	

Use Case UC-12	CancelingForm
Related Requirements	REQ-2, NONREQ-8

Initiating Actor	3rd party agents
Actor's Goal	To allow the 3rd party to close form
Participating Actors	None
Preconditions	The 3rd party agent has all required documents and must be already logged into the application.
Postconditions	Form fields are cleared and form closes
Flow of Events for Main Success Scenario: →1. Third party presses the canceling button ←2. System clearPress all field and closes the form	

Use Case UC-13	SubmitForm
Related Requirements	REQ-1, REQ-2, REQ-3
Initiating Actor	3rd party agents
Actor's Goal	To allow the system to capture all fields in the form that was filled out by the user
Participating Actors	None
Preconditions	The 3rd party agent has all required documents and must be already logged into the application.
Postconditions	The form is now completed
Flow of Events for Main Success Scenario: →1. Third party fills out each field of the form. ←2. Admin office Checks to make sure no fields are not empty. ←3. System Makes Submit button available to the user	

System Sequence Diagrams:

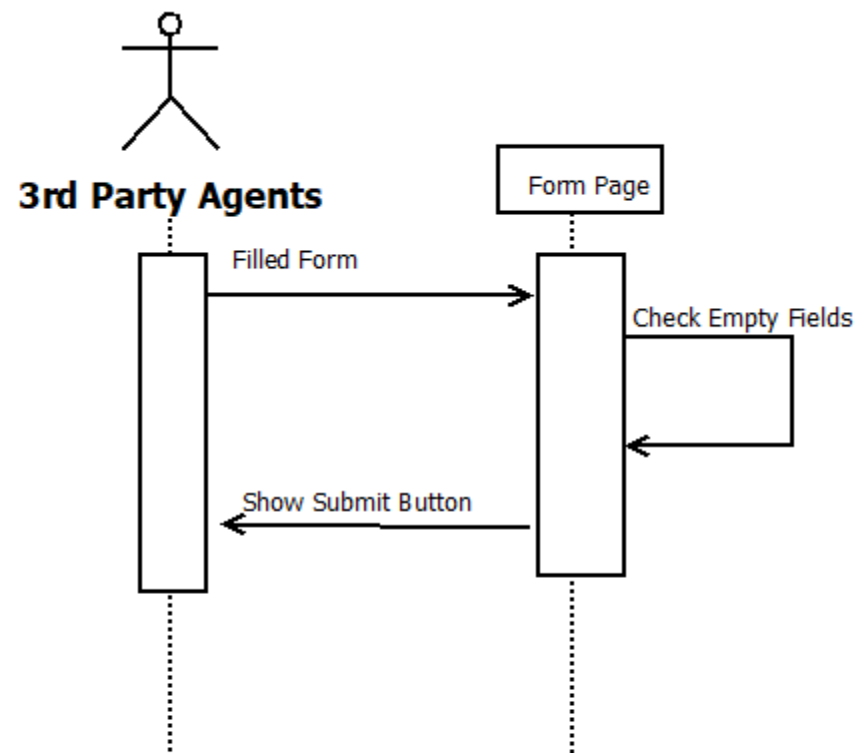


Figure x. The sequence diagram for UC-1

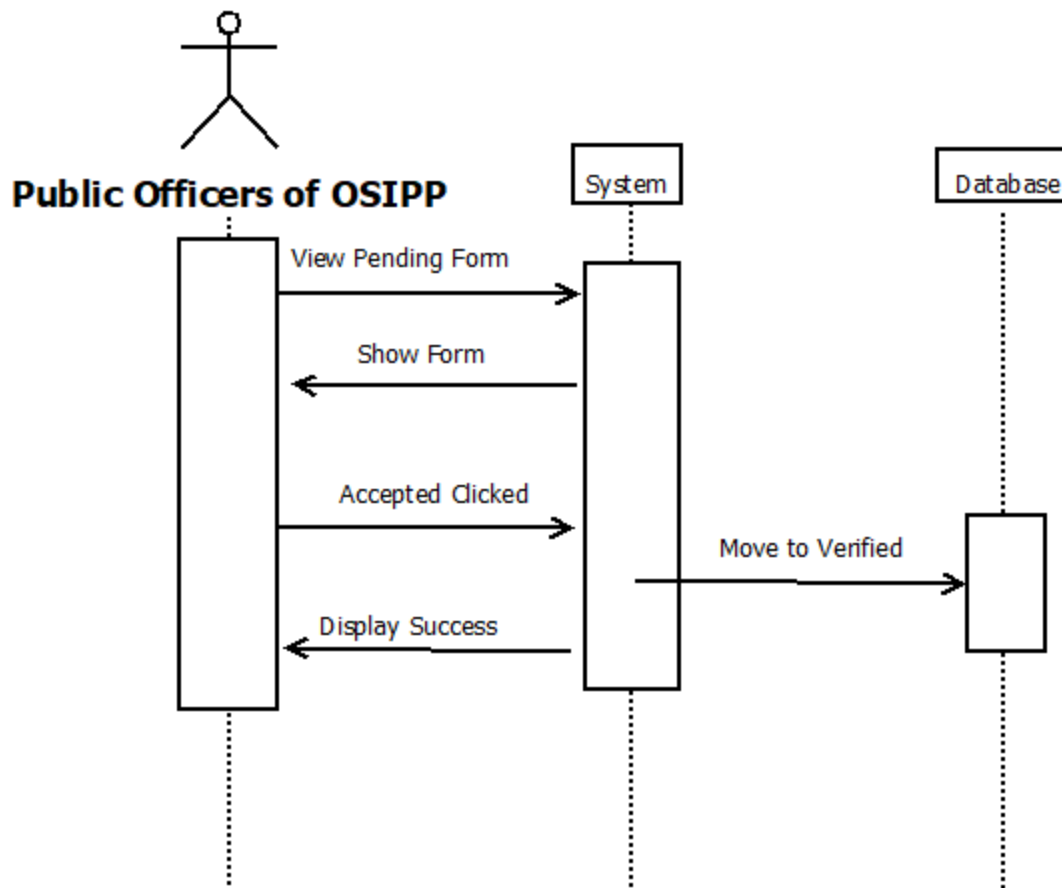


Figure 6. The sequence diagram for UC-2

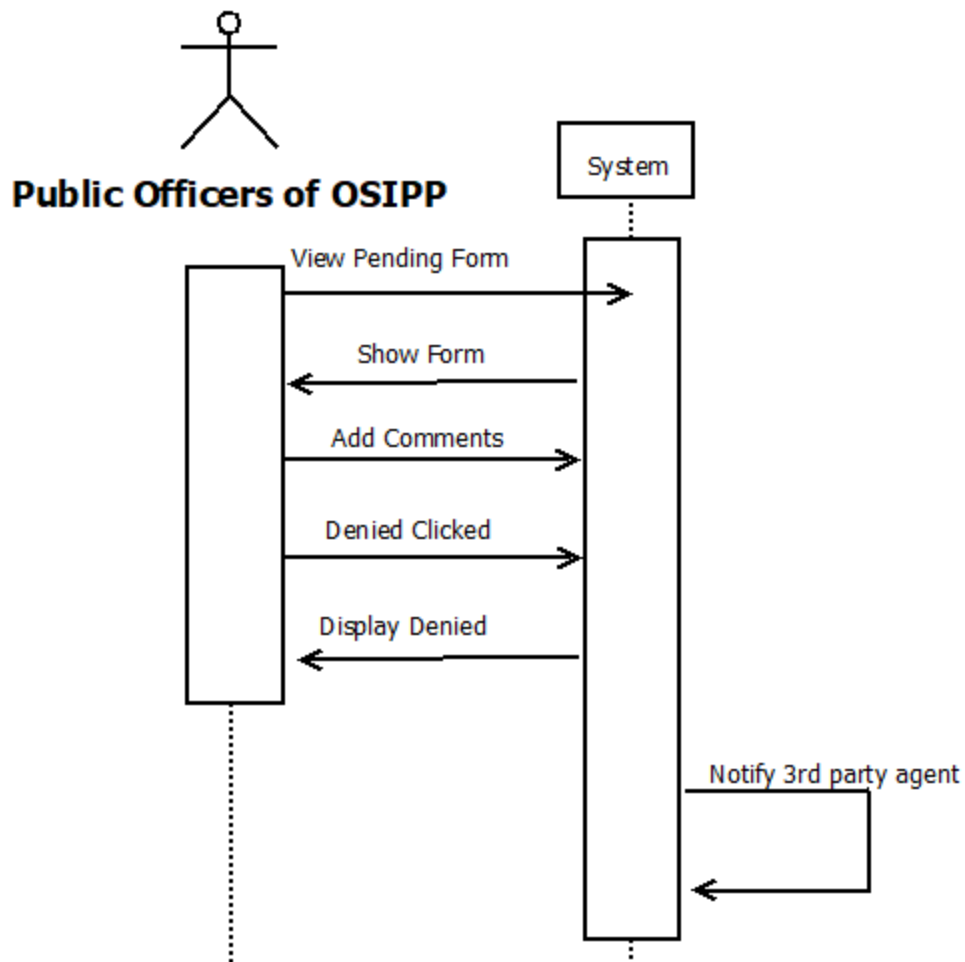


Figure 7. The sequence diagram for UC-3

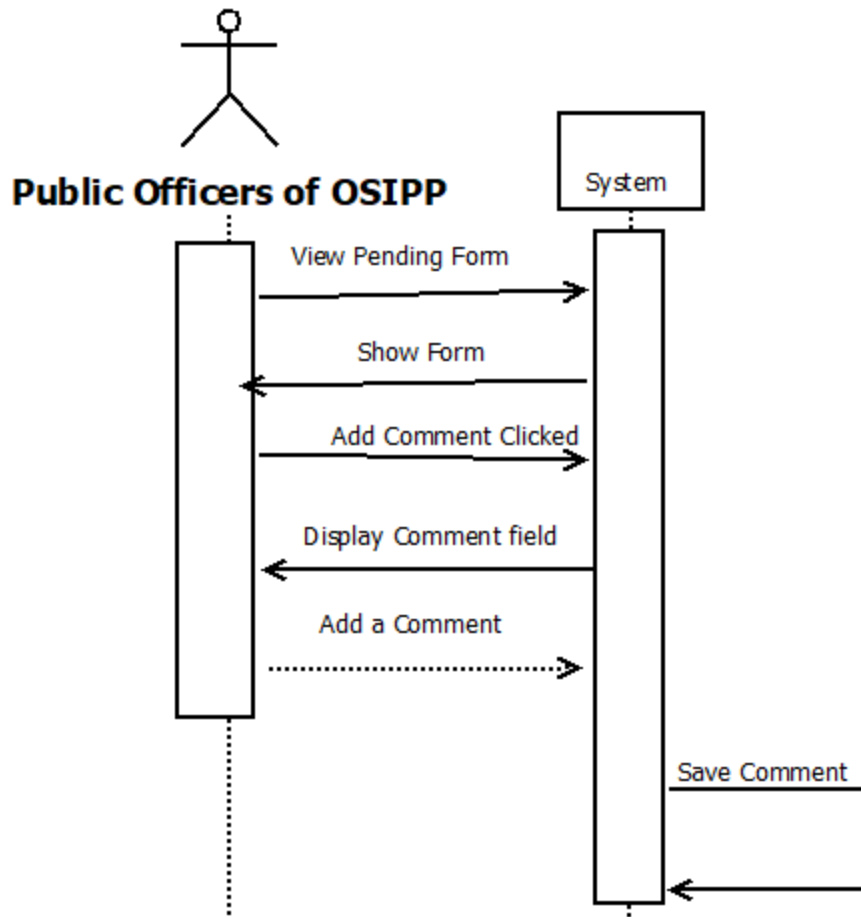


Figure 8. The sequence diagram for UC-4

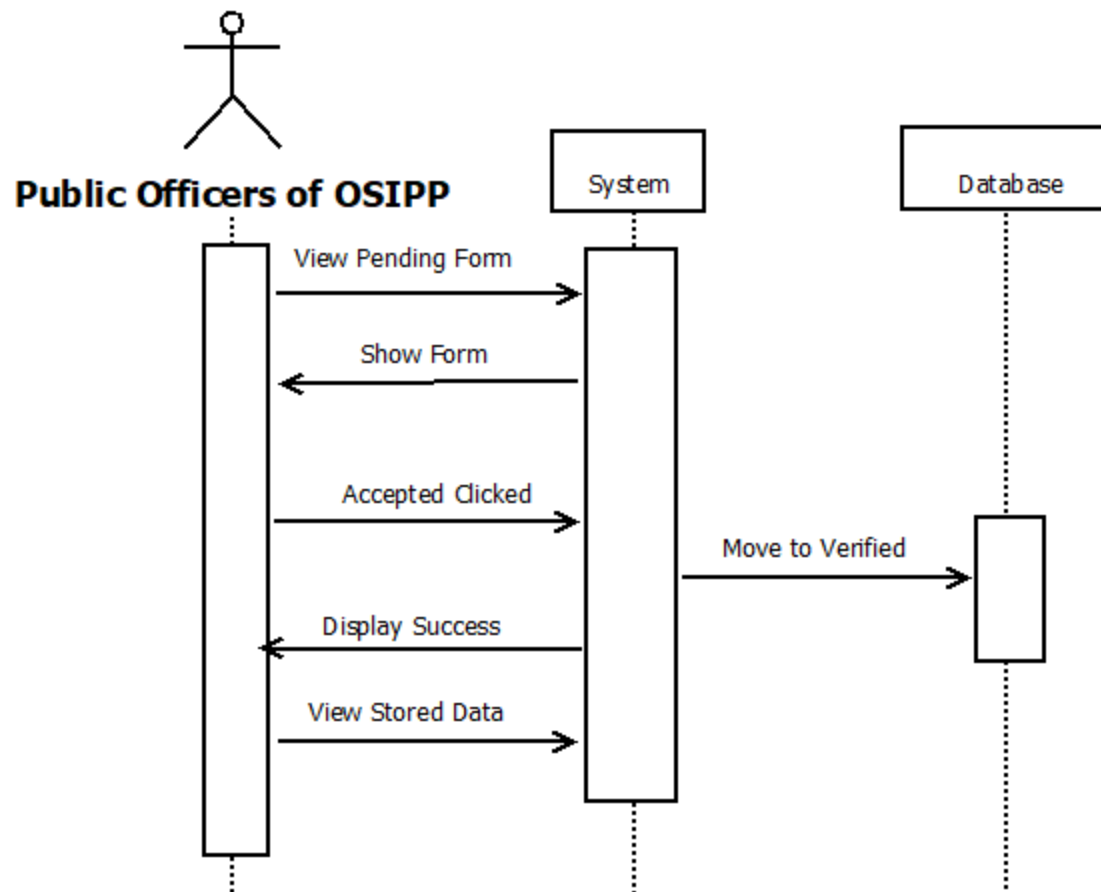


Figure 9. The sequence diagram for UC-5

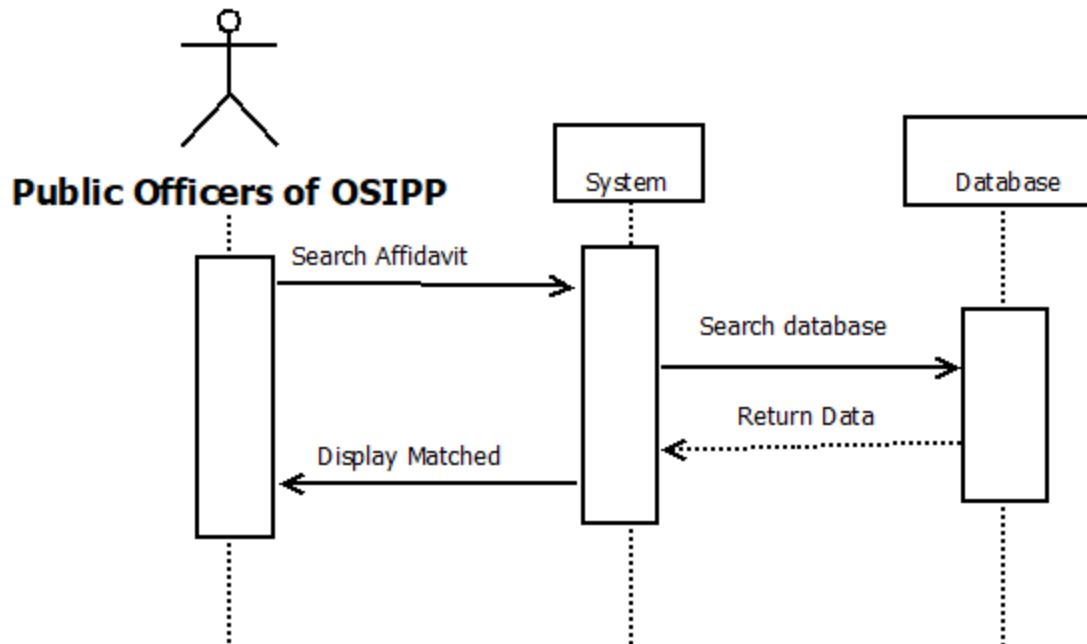


Figure 10. The sequence diagram for UC-6

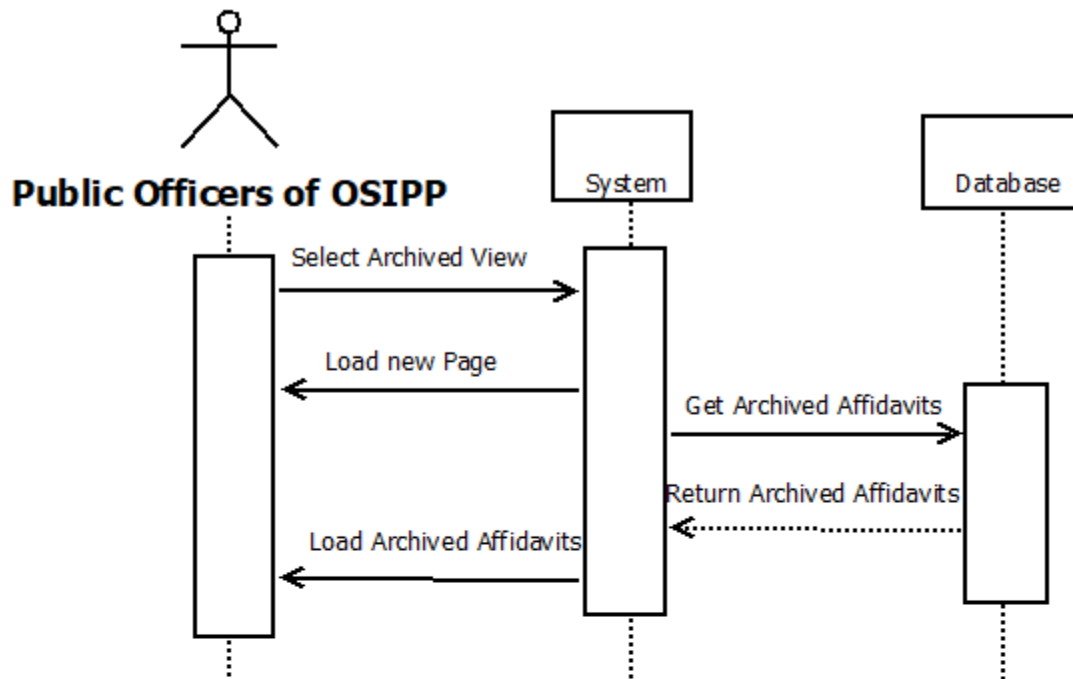


Figure 11. The sequence diagram for UC-7

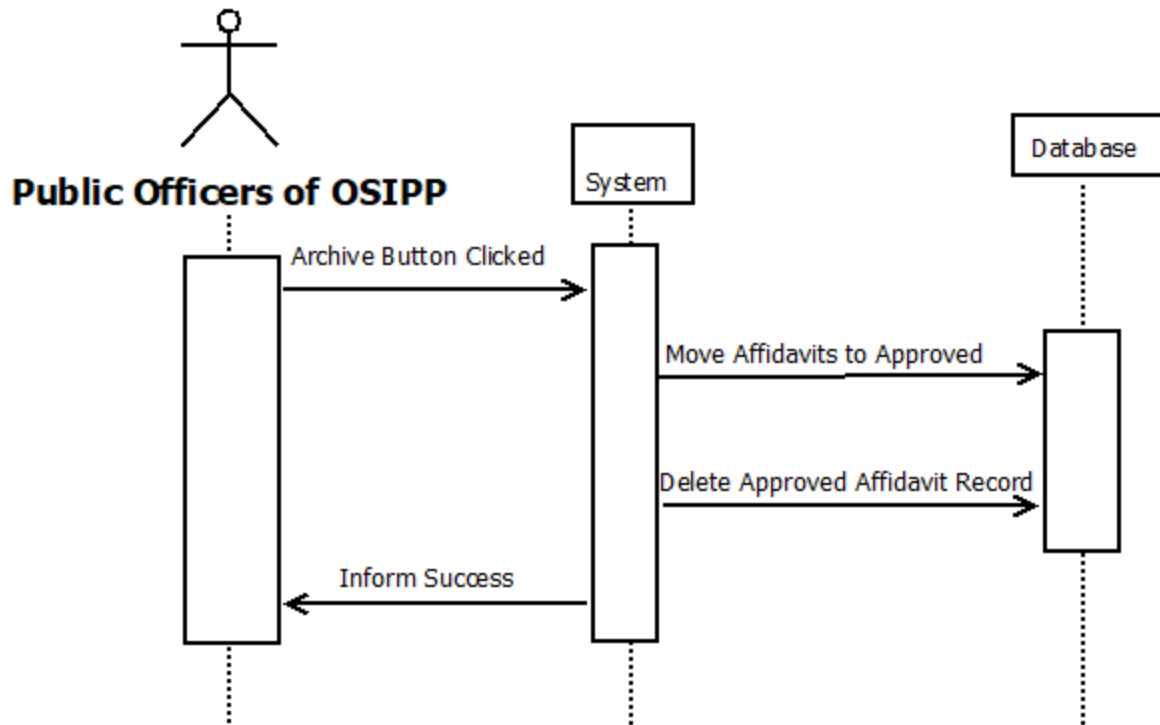


Figure 12. The sequence diagram for UC-8

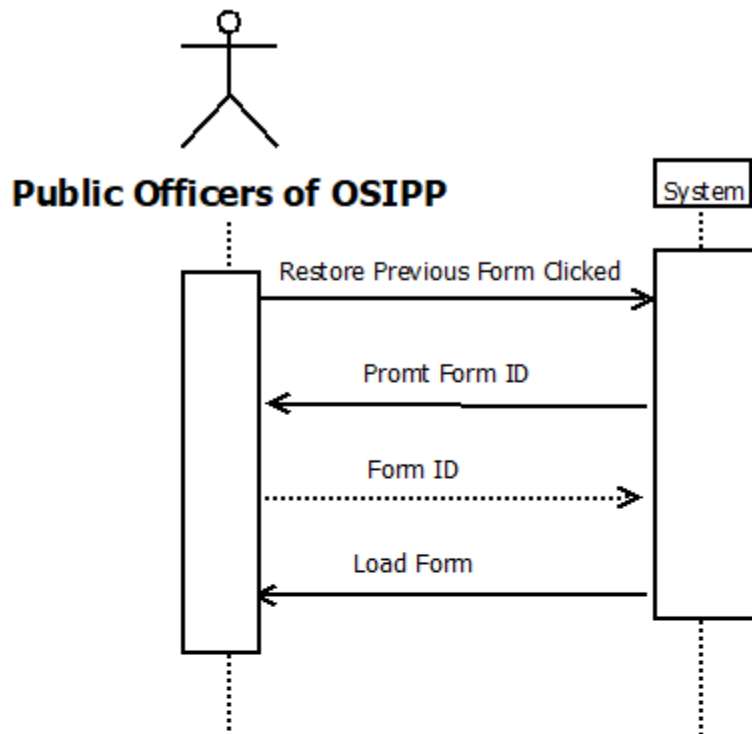


Figure 13. The sequence diagram for UC-9

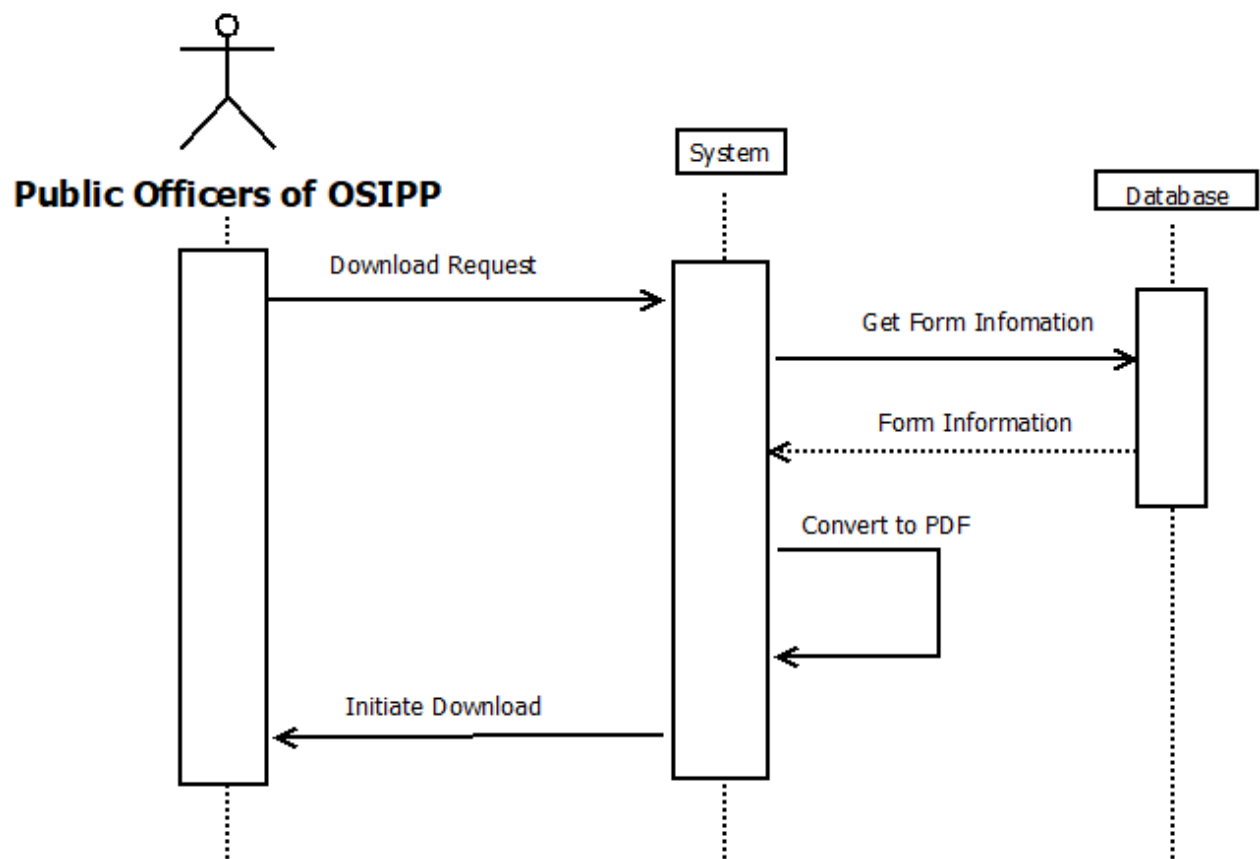


Figure 14. The sequence diagram for UC-10

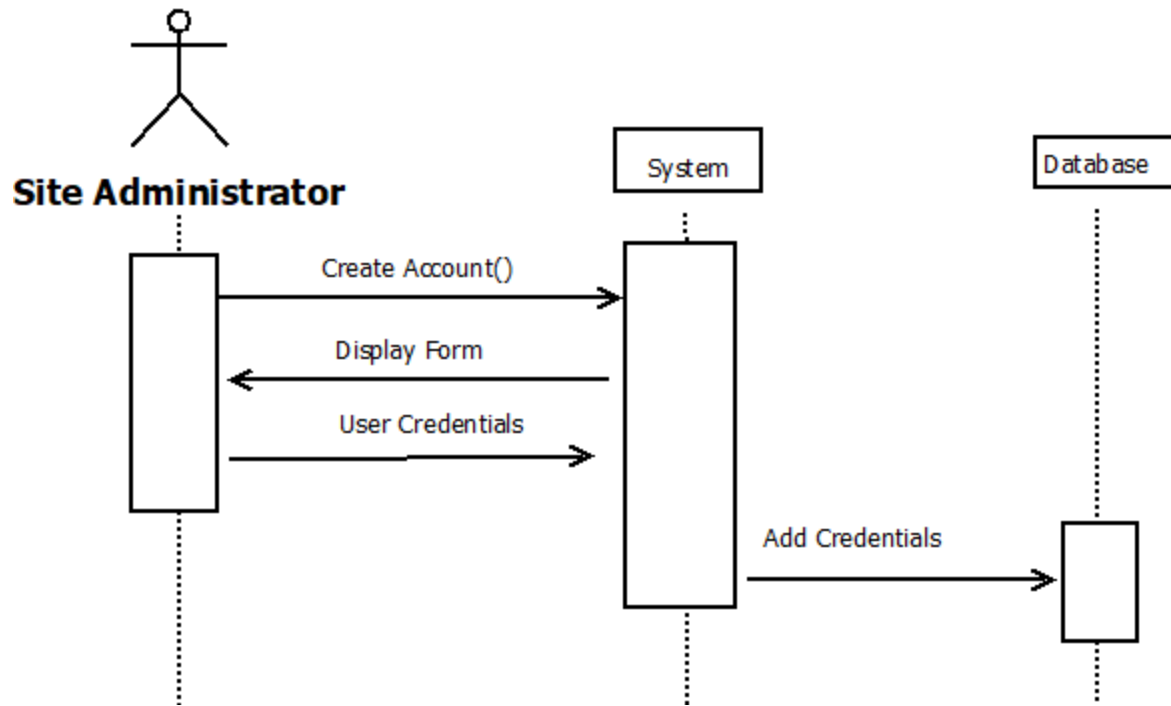


Figure 15. The sequence diagram for UC-11

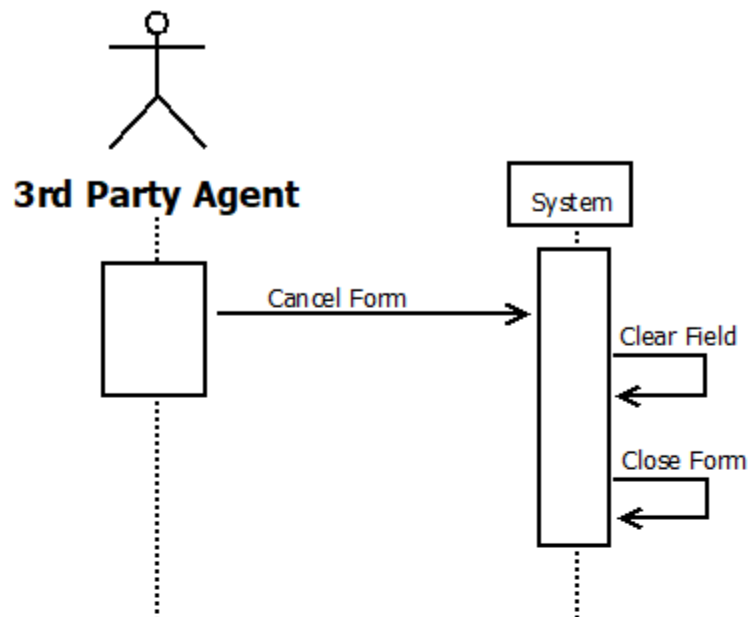


Figure 16. The sequence diagram for UC-12

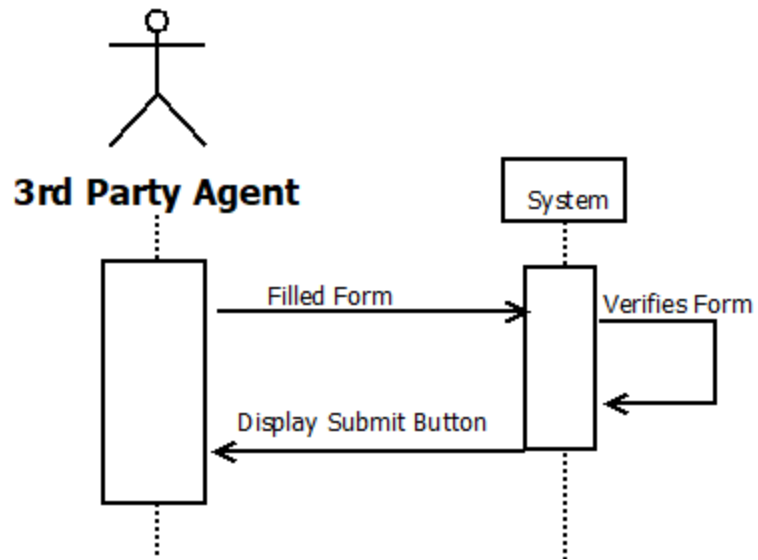


Figure 16. The sequence diagram for UC-13

EFFORT ESTIMATION

Actor Type	Simple Average Complex	Average	Complex
Number of Actors	1	2	2
Weight	1	2	3
UAW	11		

Use Cases Types	Simple Average Complex	Average	Complex
Number of Use Cases	10	2	1
Weight	1	2	3
UUCW	17		

$$\text{Unadjusted Use Case Points} = \text{UAW} + \text{UUCW} = 11 + 17 = 28$$

Technical Factor	Weight	Perceived Complexity	Calculated Factor
Distributed system	2	3	6
Performance Objectives	2	3	6
End-user Efficiency	1	3	3
Complex Processing	1	2	2
Reusable Code	1	1	1
Easy to Install	0.5	2	1
Easy to use	0.5	2	1
Portable	2	3	6
Easy to change	1	2	2
Concurrent	1	2	2
Security Features	1	4	4
Access to Third Parties	1	2	2

Training needs	1	0	0
Technical Factors Total	36		
Technical Complexity Factor	0.96		

Environmental Factors	Weight	Calculated Factors	Impact
Familiar with the development process	1.5	3	4.5
Application Experience	0.5	3	1.5
Object Oriented Experience	1	3	3
Lead Analyst capability	0.5	2	1
Motivation	1	4	4
Stable requirements	2	1	2

Part-time staff	-1	0	0
Difficult Programming Language	-1	1	-1

Efactor = 15

Environment Complexity Factor = 0.95

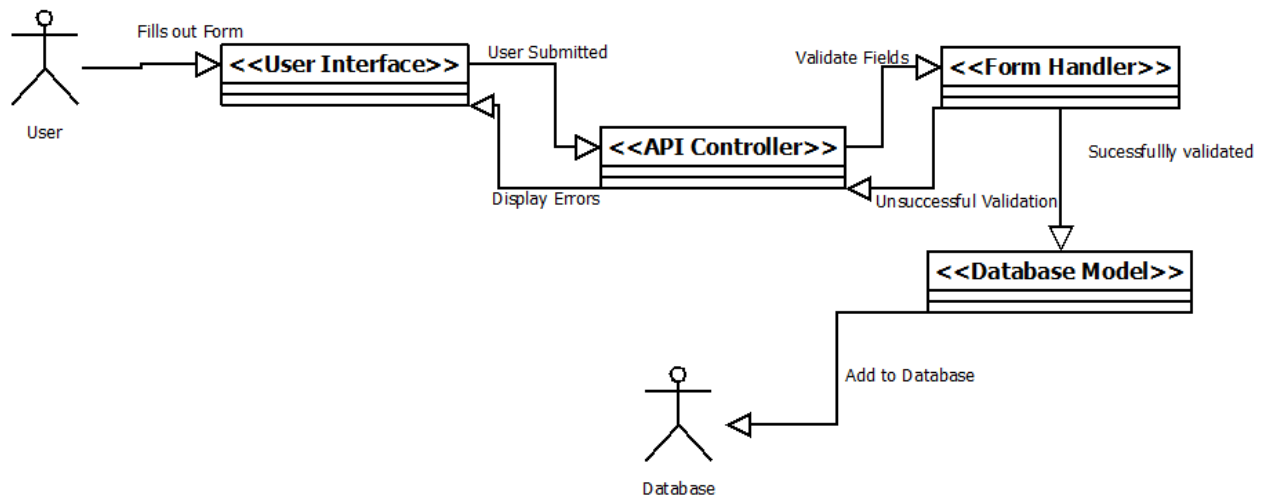
Adjusted Use Case Points = $28 * 0.96 * 0.95 = 25.53$

Staff hour per UCPoints = 28

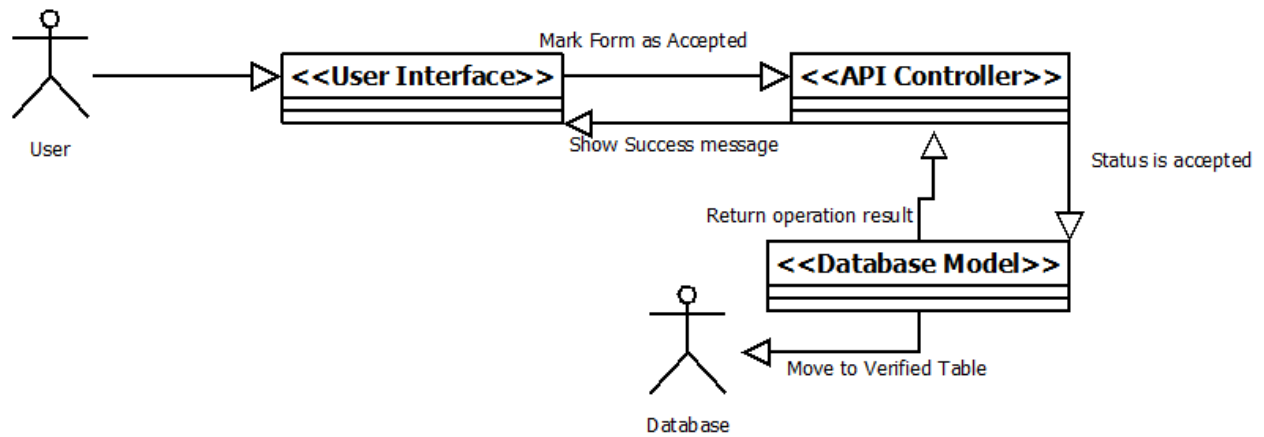
Effort Estimate = $25.53 * 28 = 715$ hrs

DOMAIN ANALYSIS

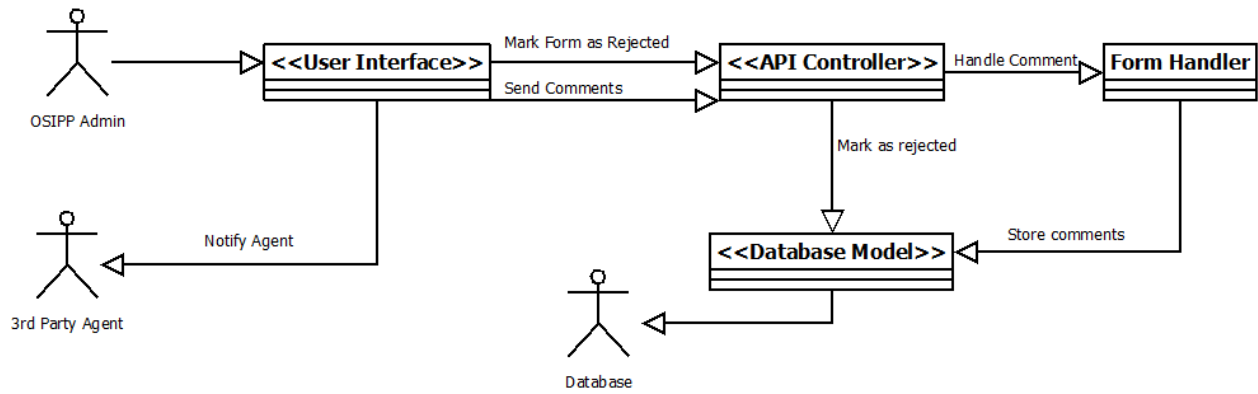
Domain Model



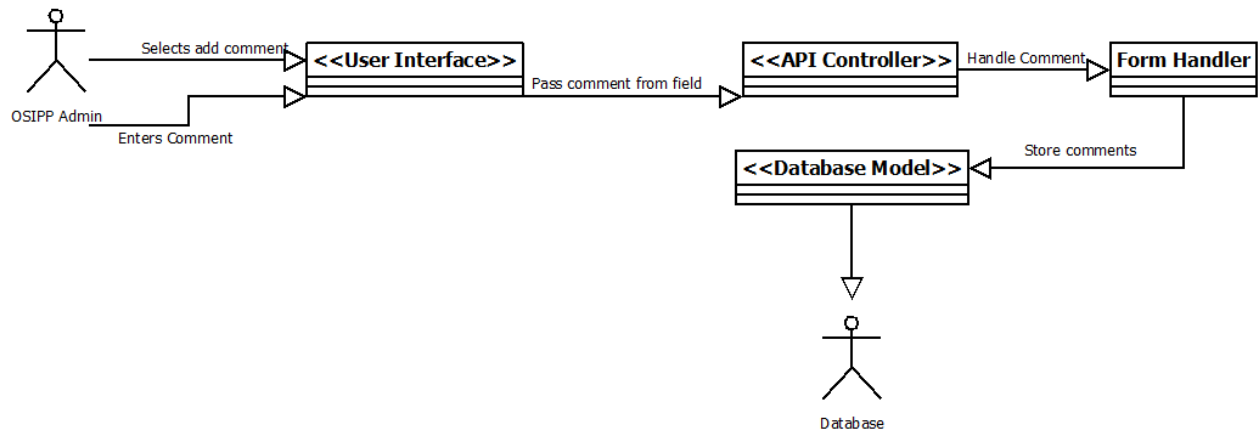
UC 1 - FillingForm



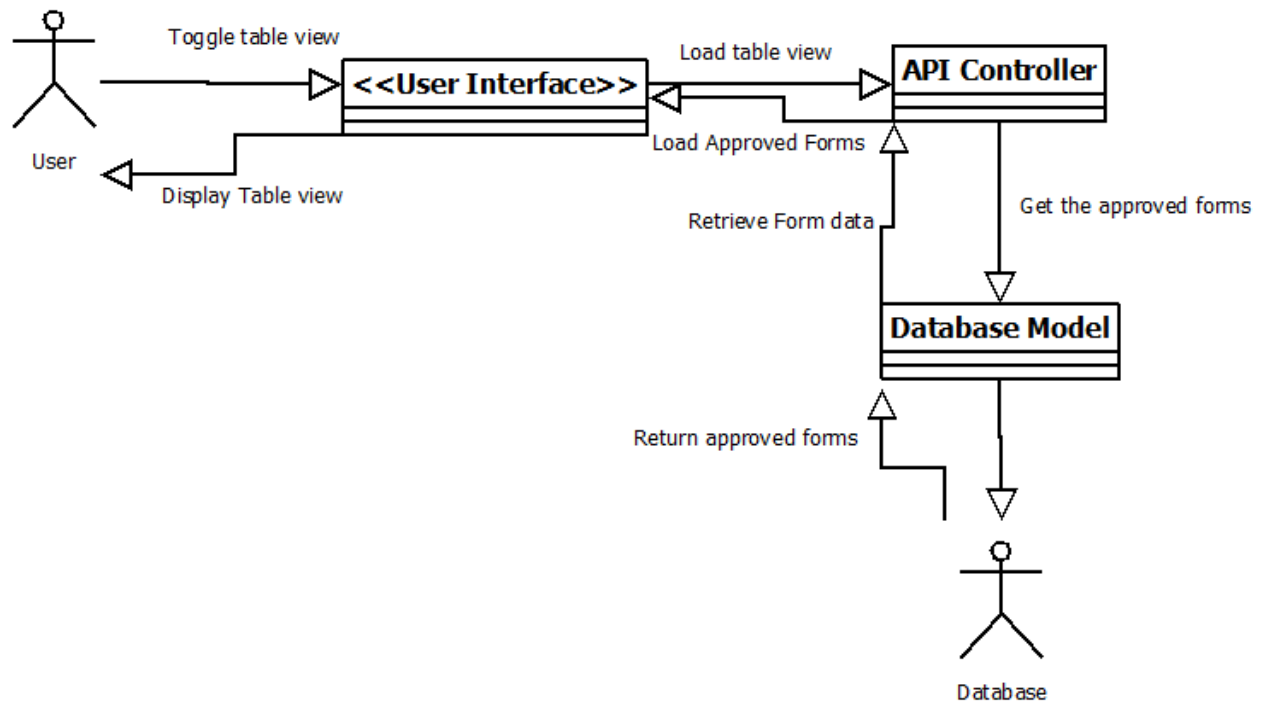
UC 2 - AcceptingForm



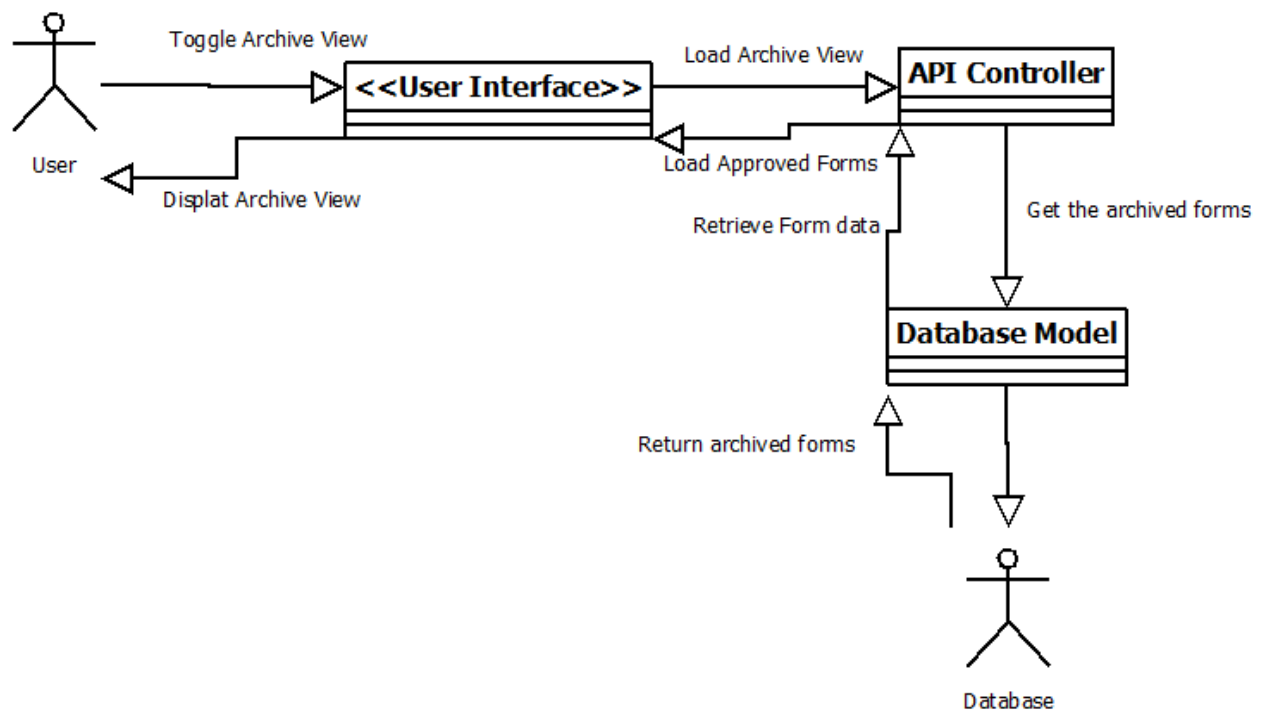
UC 3 - DenyingForm



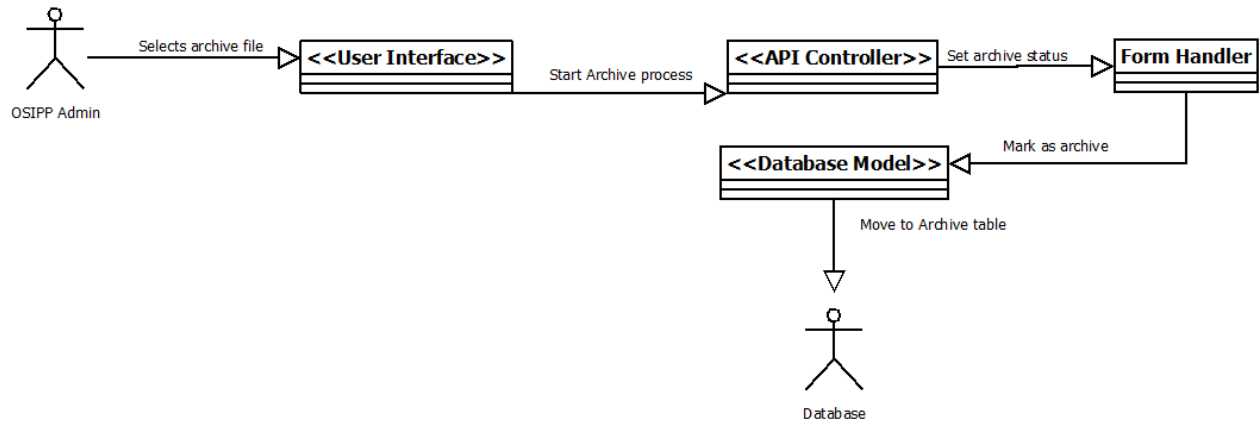
UC 4 - AddingComments



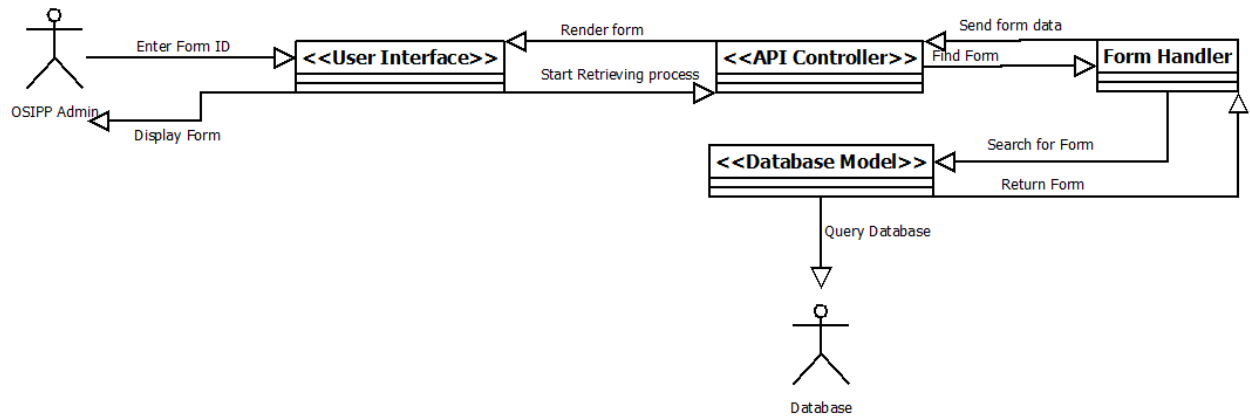
UC 6 - ToggleTableView



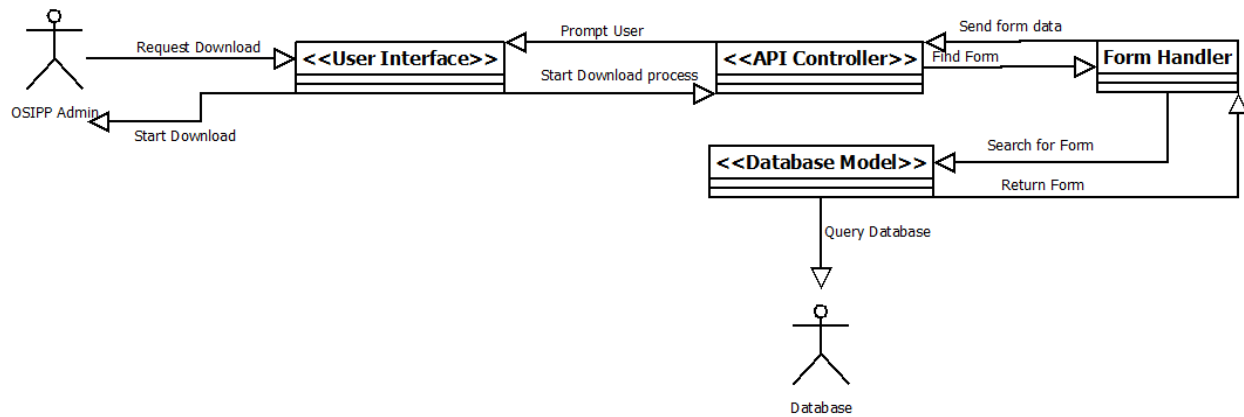
UC 7 - Archive View



UC 8 - Archiving file



UC 9 - Restoring Form



UC 10 - Downloading Form

Concept Definition

Responsibility Description	Concept Name
Provide the necessary fields to fill out the form from the user.	User Interface
Will delegate the task to another handler. Will generate error messages to the user	API Controller
Responsible for validating the user input. Forward the reason for unsuccessful validation	Form Handler
Add the form to the pending table.	Database Model

Concept Definition for UC-1

Responsibility Description	Concept Name
Provide the graphical element for the user to mark the form as accepted Display the success message to the user	User Interface
Contacting the database to mark as accepted	API Controller

Return the success message to the UI	
Move the form to the verified table	Database Model

Concept Definition for UC-2

Responsibility Description	Concept Name
Provides the field to input comments	User Interface
Receive the comments Delegate to the form handler set up to mark form as rejected	API Controller
Process comments	Form Handler
Mark the form as rejected Store the comments for the form	Database Model

Concept Definition for UC-3

Responsibility Description	Concept Name
Provide a method for users to add and enter comments	User Interface
Delegate to the form handler	API Controller
Communicate with the database to store comments	Form Handler
Save the comments in the database	Database Model

Concept Definition for UC-4

Responsibility Description	Concept Name
Allow the user to toggle table view. The UI will display the table view	User Interface
Communicate with the database to retrieve all approved forms. Send approved forms to the UI	API Controller

Send approved form data to the API Controller	Database Model
---	----------------

Concept Definition for UC-6

Responsibility Description	Concept Name
Allow the user to archived view. The UI will display the archived view	User Interface
Communicate with the database to retrieve all archived forms. Send archived forms to the UI	API Controller
Send archived form data to the API Controller	Database Model

Concept Definition for UC-7

Responsibility Description	Concept Name
Provide the UI t the user to archive the form	User Interface
Delegate to the form handler	API Controller
Communicate with the database model to mark as archive	Form Handler
Move form to archived table	Database Model

Concept Definition for UC-8

Responsibility Description	Concept Name
Provide the interface to enter form ID Displays the result of the form ID	User Interface
Start the retrieving process Sending form data to be processed by the UI	API Controller
Requesting the form data from the database	Form Handler

Forward the response to the controller	
Search database for form Return form to the handler	Database Model

Concept Definition for UC-9

Responsibility Description	Concept Name
Allow the user to start the download process Start the download	User Interface
Start the retrieving process Sending form data to be processed by the UI	API Controller
Requesting the form data from the database Forward the response to the controller	Form Handler
Search database for form Return form to the handler	Database Model

Concept Definition for UC-10

Association Definition:

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	API Controller handles the front end input. API sends error to the front end	Receive
API Controller ↔ Form Handler	Validate the form	Operation
Form Handler ↔ Database Model	Sends the user data to the database	Process Data

Association Definition for UC-1

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Receive a signal to mark the form as accepted	Receive

API Controller ↔ Database Model	Move the data from the user to the database	Process Data
---------------------------------	---	--------------

Association Definition for UC-2

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Receive feedback to reject the form	Receive
API Controller ↔ Form Handler	Process any comment made	Process
Form Handler ↔ Database Model	Sends any comment to the database	Process Data
API Controller ↔ Database Model	Marks the form as rejected	Operation

Association Definition for UC-3

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Retrieve comments left	Retrieve
API Controller ↔ Form Handler	Process any comment made	Process
Form Handler ↔ Database Model	Sends any comment to the database	Process Data

Association Definition for UC-4

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Receive a request to toggle table view	Receive
API Controller ↔ Database Model	Query the database for appropriate form	Operation

Association Definition for UC-6

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Receive a request to toggle archive view	Receive
API Controller ↔ Database Model	Query the database for appropriate form	Operation

Association Definition for UC-7

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Receive a request to archive a form	Receive
API Controller ↔ Form Handler	Set the status of the form	Operation
Form Handler ↔ Database Model	Move the form data to the archive table	Process Data

Association Definition for UC-8

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Receive a request to retrieve the form. API sends the form data to the UI	Receive
API Controller ↔ Form Handler	Request the form data	Operation
Form Handler ↔ Database Model	Retrieve the data from the database.	Process Data

Association Definition for UC-9

Concept Pair	Association Definition	Association Name
User Interface ↔ API Controller	Receive a request to download form. Prompt the download sequence	Receive

API Controller ↔ Form Handler	Request the form data	Operation
Form Handler ↔ Database Model	Retrieve the data from the database. Database send form data to fulfill request	Process Data

Association Definition for UC-10

Attribute Definitions:

Attribute Definitions for UC-1 FillingForm, UC-2 AcceptingForm, UC-3 DenyingForm, UC-6 ToggleTableView, UC-7 ArchiveView, UC-8 Archiving File, UC-9 RestoringForm, UC-10 Downloading Form

Concept	Attributes	Attribute Description
User Interface	formData	All information necessary for the completion of the biographical affidavit of the agent interacting with the applying
Database Model	ModelData	A tabulated version of the form information

Attribute Definition for UC-4 AddingComments

Concept	Attributes	Attribute Description
User Interface	comments	Notes left behind on specific form entries by compliance officers at OSSIP.
Database Model	commentData	A database-friendly version of each comment stored separately from the modelData

System Operation Contract:

Use Case	UC-1: FillingForm
Contract Name	FillForm(UserInfo)
Responsibilities	Fill all fields of form
Preconditions	Access form
Postconditions	Verify each field was filled

Use Case	UC-2: AcceptingForm
Contract Name	AcceptForm()
Responsibilities	All fields of form filled
Preconditions	Submit form
Postconditions	Added to database

Use Case	UC-3: DenyingForm
Contract Name	DenyForm()
Responsibilities	Reject form once information entered was incorrect
Preconditions	Submit form
Postconditions	Return form to user

Use Case	UC-4: AddingComments
Contract Name	AddComment()
Responsibilities	Add comments to invalid fields of the form during the verification phase
Preconditions	Submit Form
Postconditions	Return to user

Use Case	UC-5: StoringData
Contract Name	StoreData()
Responsibilities	Add data to database
Preconditions	Successful Verification
Postconditions	Add to database

Use Case	UC-6: TogglingTableView
Contract Name	ToggleView()
Responsibilities	Display all approved forms
Preconditions	Add to database
Postconditions	Toggle view

Use Case	UC-7: CancelingForm
Contract Name	CancelForm()
Responsibilities	cancel form and clear all fields.
Preconditions	Open form
Postconditions	Close form

Use Case	UC-8: Submitting form
Contract Name	SubmitForm()
Responsibilities	Submit the form after properly being filled out
Preconditions	Fill all fields
Postconditions	Verification

Use Case	UC-9: ViewArchive
Contract Name	ViewArchive()
Responsibilities	View all archive records
Preconditions	Add to database
Postconditions	View records

Use Case	UC-10: ArchivingFile
Contract Name	ArchiveFile()
Responsibilities	Archive a file during the verification phase
Preconditions	Add to database
Postconditions	View Archive

Use Case	UC-11: RestoringForm
Contract Name	RestoreForm()
Responsibilities	Restore a form to user using a unique restoration code
Preconditions	Submit form
Postconditions	Edit invalid fields

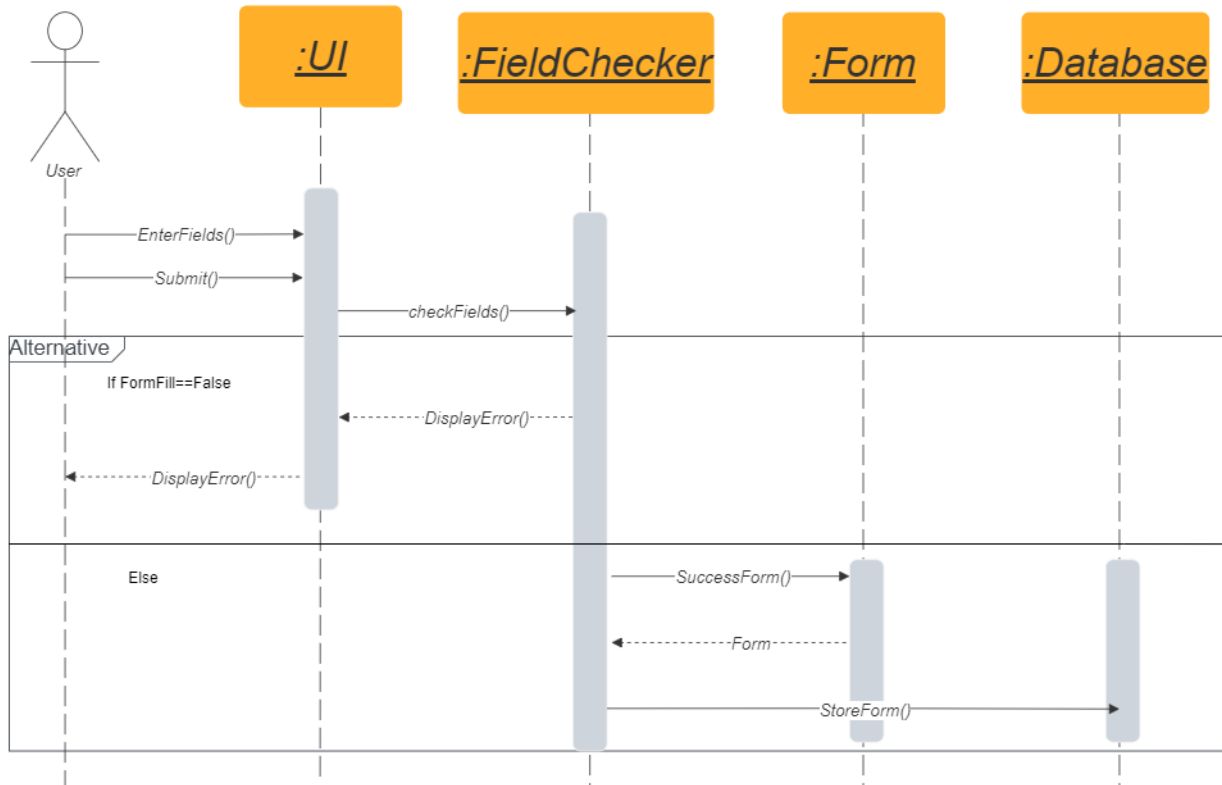
Use Case	UC-12: DownloadForm
Contract Name	Downloadform()
Responsibilities	Downloaded form into a PDF format
Preconditions	Add to database
Postconditions	View download

Use Case	UC-13: MakingAccount
Contract Name	MakeAccount()
Responsibilities	Create account for OSIPP public officers
Preconditions	GOB email address
Postconditions	Login Credentials

INTERACTION DIAGRAMS

Diagrams:

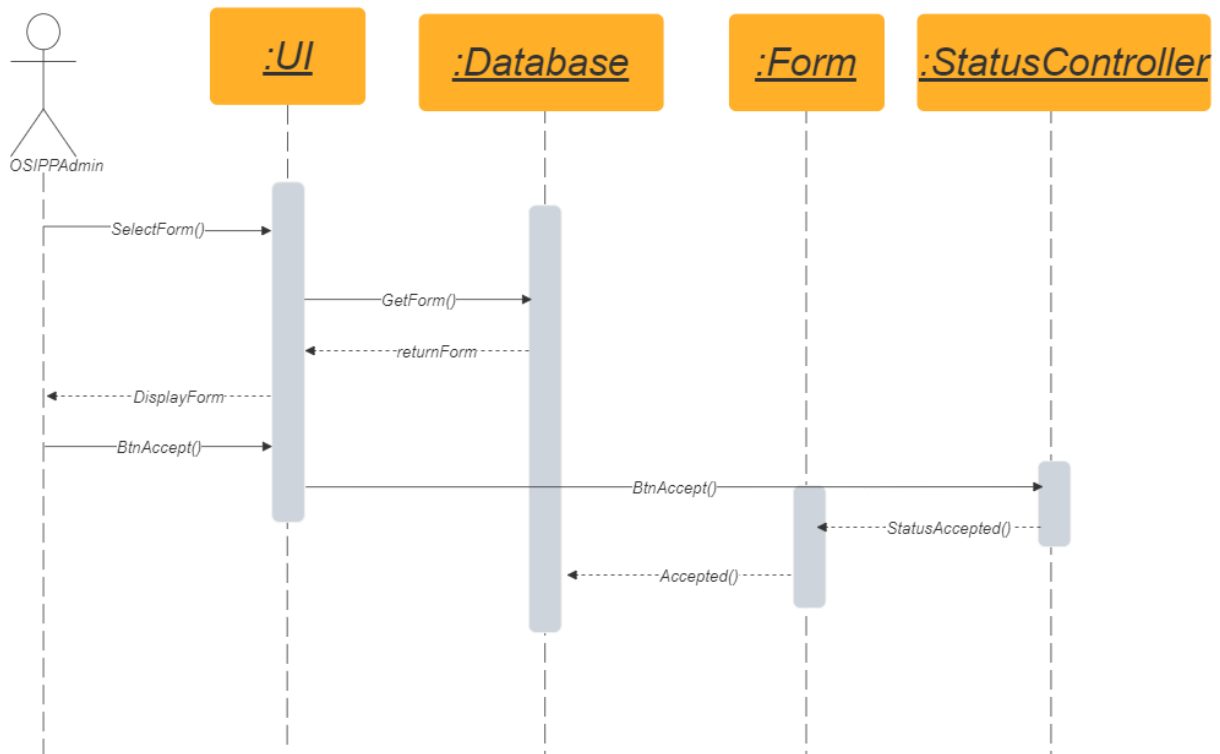
UC1: FillingForm



Description for UC-1:

Function EnterFields will allow the 3rd party user to enter the necessary information within the UI of the form and user clicks the submit button, FIELDCHECKER ensures that the data entered on the form is valid. If the data entered was incorrect DisplayError will be tasked to notify the user letting them know what needs to be fixed. StoreForm will allow the form to be stored onto the temp DATABASE for verification.

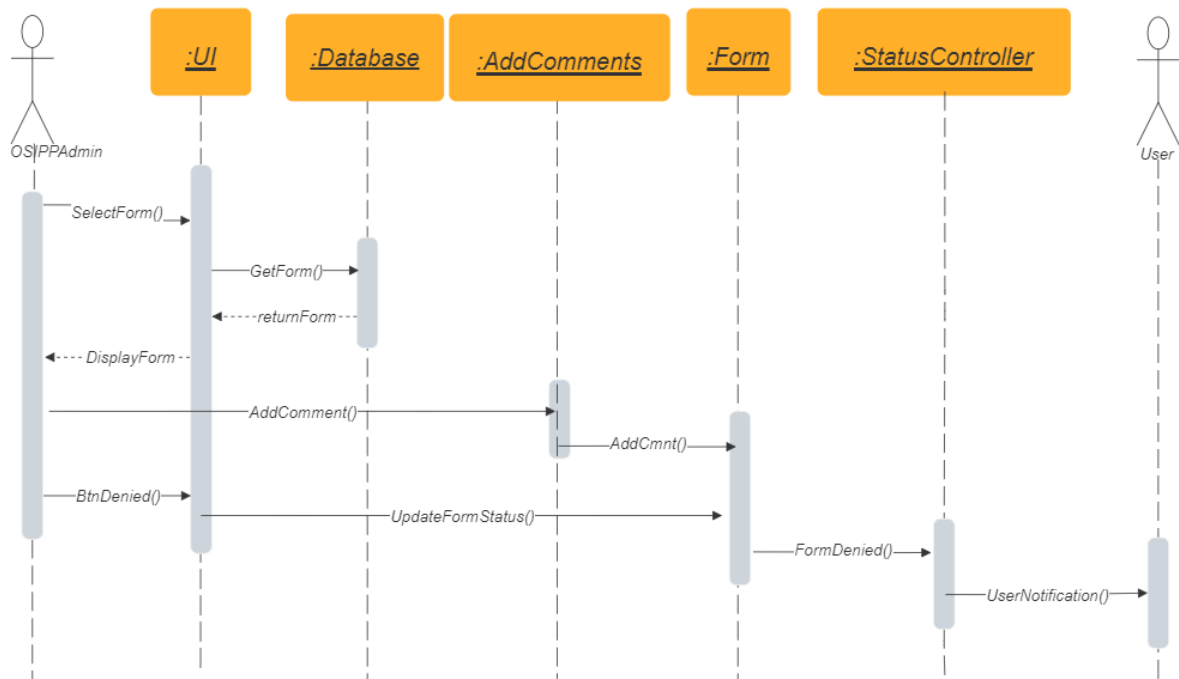
UC2: Accept Form



Description for UC-2:

OSIPP Admin that is logged in, selects a form that shows its status as “pending”. The DATABASE allows the officer to get the form and the UI displays it in verification mode. From the main screen the user will select the Accept within the UI as long as the form was completed properly. The UI sends the accepted form to the database, then the STATUSCONTROLLER will send a request to the Form thus changing its status from pending to accepted.

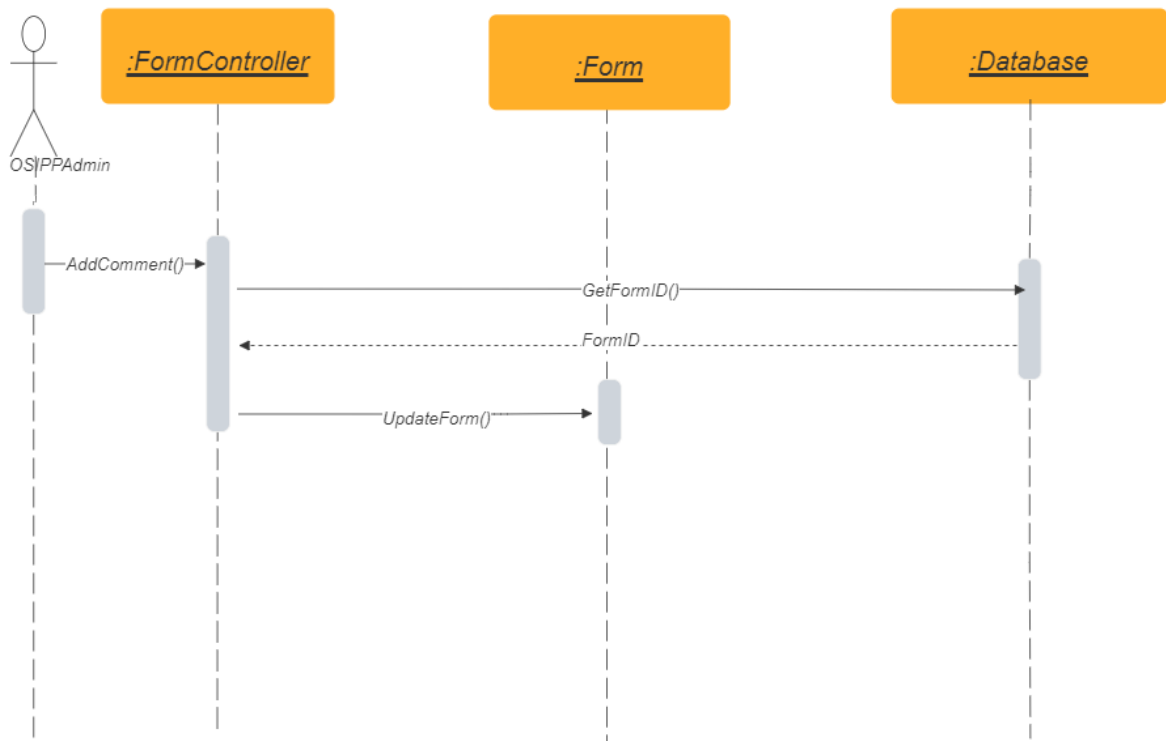
UC3: Deny Form



Description for UC-3:

SelectForm allows the OSIPP Admin to choose and select a form within the UI that has shown its status as “pending”. The DATABASE allows the officer to get the form and the UI displays it in verification mode. The OSIPP Admin clicks on the ADDCOMMENTS button on the main screen to allow the user to construe each field of the form that the user made a mistake. The user clicks on the deny button then the STATUSCONTROLLER sends a request to the form that its status changed to “unverified” and notifies the user about the necessary field highlighted that needs to be fixed.

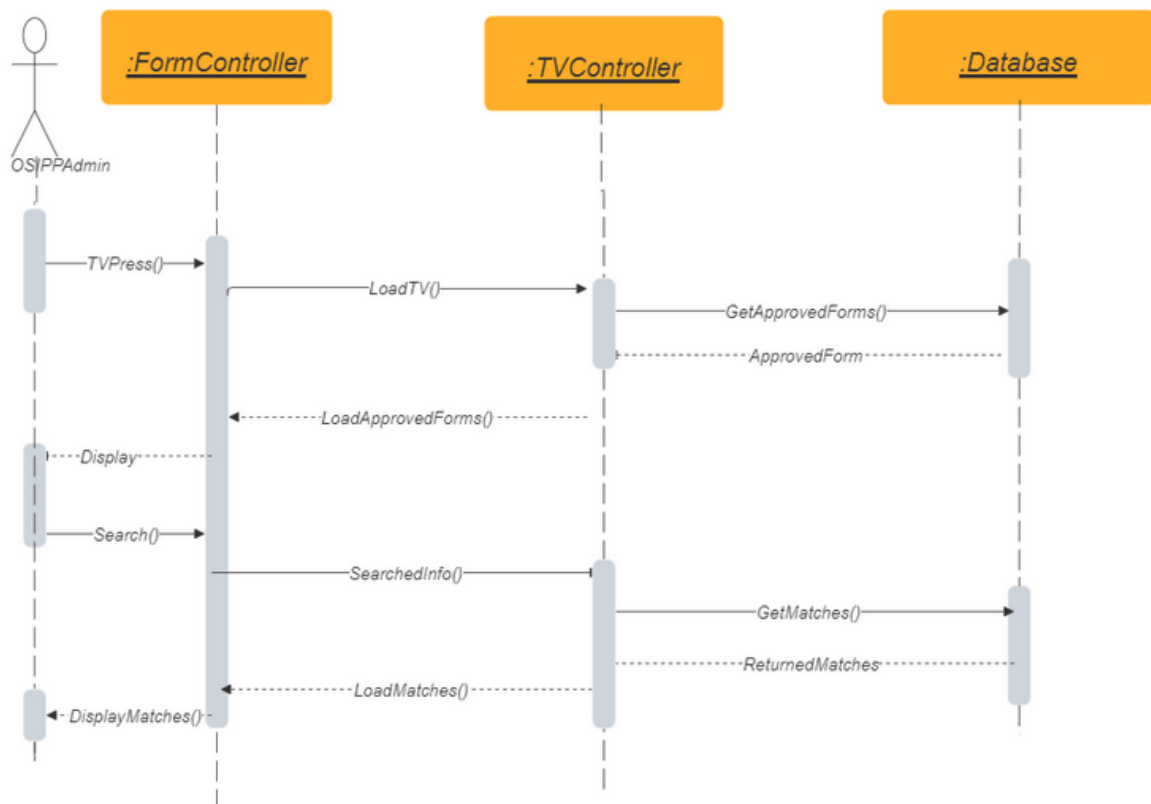
UC4: Add Comments



Description for UC-4:

Once the OSIPPAAdmin makes the comments to the form, the FORMCONTROLLER sends a request to the DATABASE to fetch the ID of the form and to return it back to the controller. Lastly, the FORMCONTROLLER will send the request to have the form updated with the comments added from the OSIPPAAdmin.

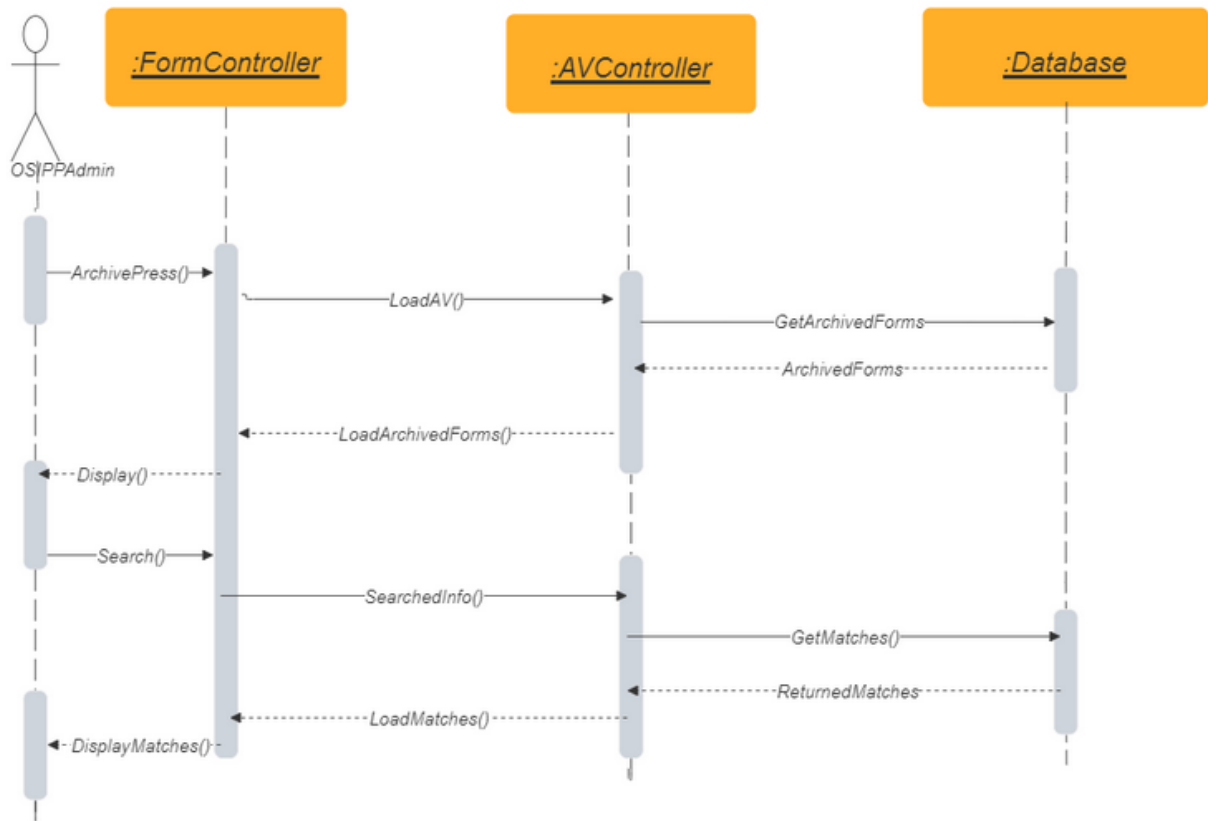
UC6: Toggling TableView



Description for UC-6:

The FORMCONTROLLER sends a request from the OSIPPAAdmin to the TVCONTROLLER to get all forms that were approved from the DATABASE. Then the DATABASE sends the request back to TVCONTROLLER, thus calling LoadApprovedForms to have only approved forms to be displayed. The OSIPP sends the search request to the FORMCONTROLLER then the TVCONTROLLER receives the request to be sent to the DATABASE to get the matched result. The request sends the ReturnMatches back to the TVCONTROLLER then to the FORMCONTROLLER to display the result of the query.

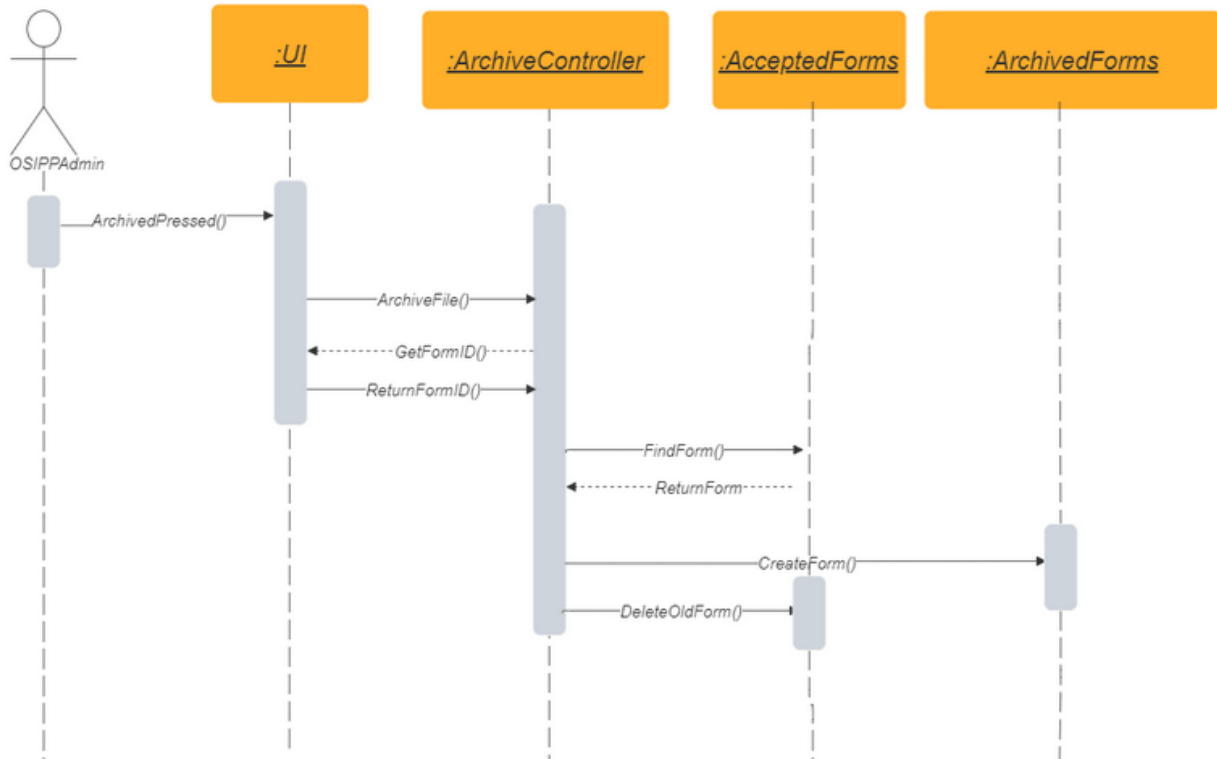
UC7: Archive View



Description for UC-7:

OSIPAdmin clicks the archive tab on the UI and the FORMCONTROLLER sends the request to the AVCONTROLLER in order for the database to get all archived forms. The request will be sent back to the AVCONTROLLER and back to the FORMCONTROLLER to have all archived forms displayed. The OSIPP sends the search request to the FORMCONTROLLER then the AVCONTROLLER receives the request to be sent to the DATABASE to get the matched result. The request sends the ReturnMatches back to the AVCONTROLLER then to the FORMCONTROLLER to display the result of the query.

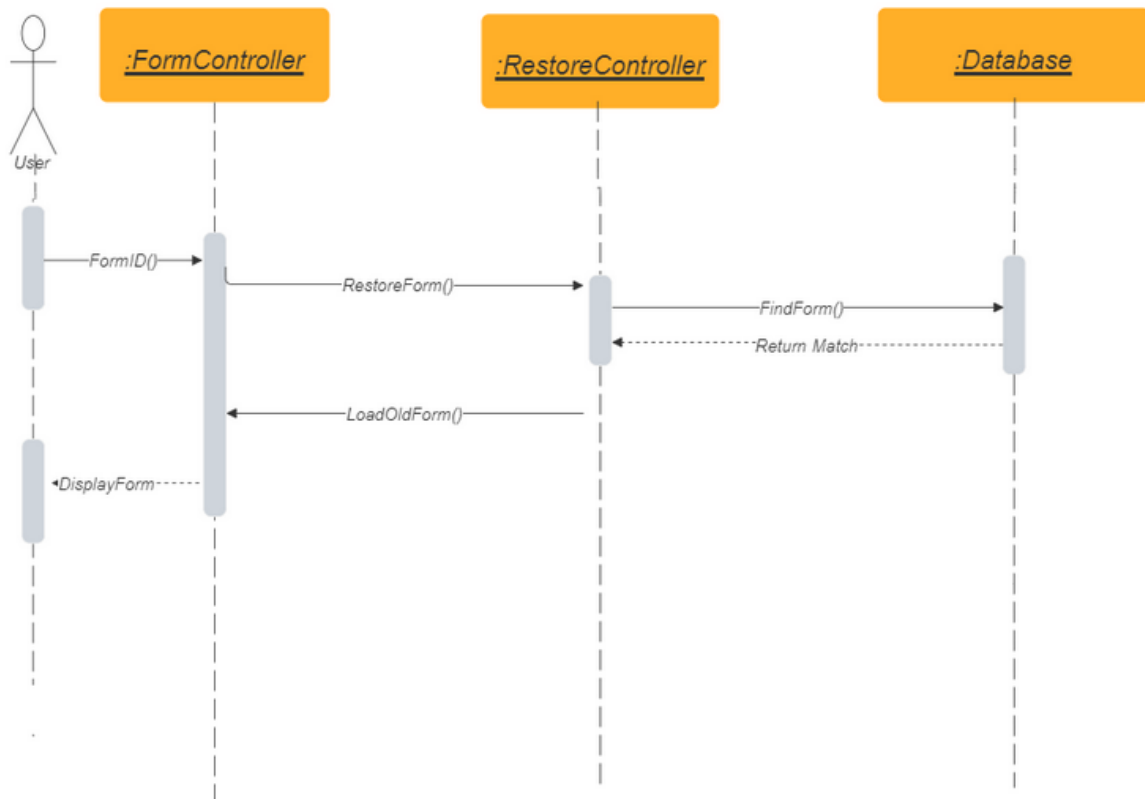
UC8: Archiving File



Description for UC-8:

OSIPPAAdmin clicks on the archive button within the main page of the UI, then ARCHIVECONTROLLER receives the request to archive the form and returns its FormID that was obtained from the UI. The ARCHIVECONTROLLER locates the form to be archived and the controller deletes the old form if requested.

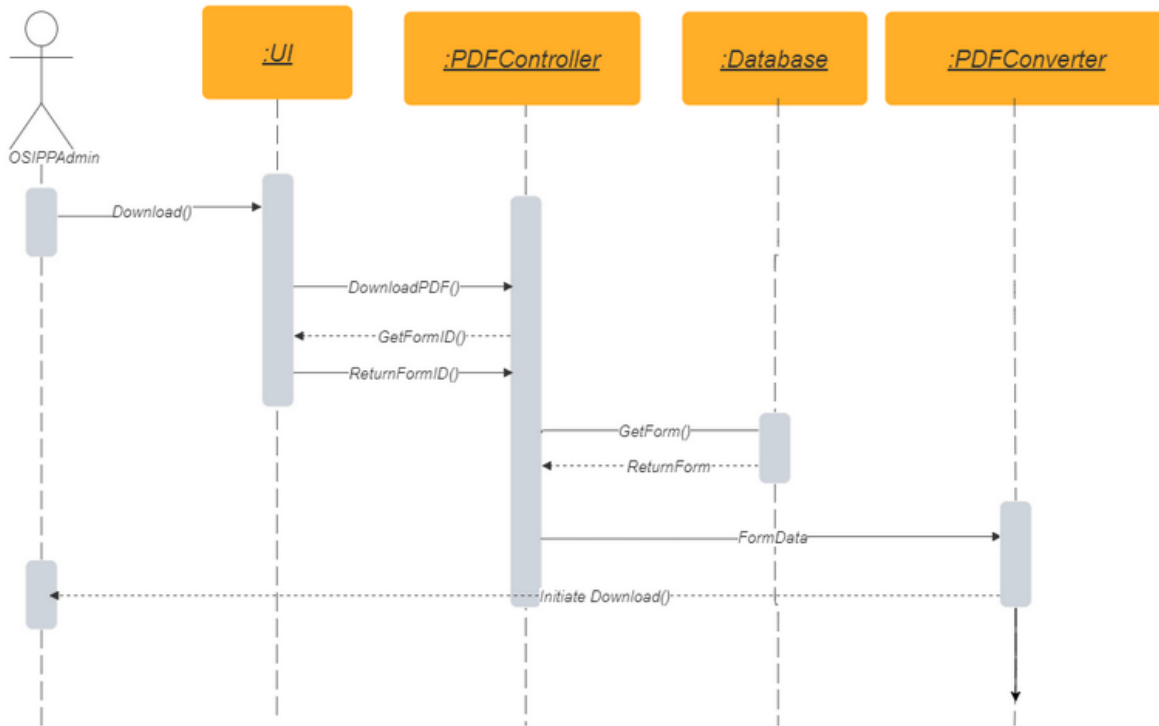
UC9: Restoring Form



Description for UC-9:

The FORMCONTROLLER sends the request to the RESTORECONTROLLER to find the form with its ID. Once the form is located from the DATABASE the result of the query is sent to the RESTORECONTROLLER then the FORMCONTROLLER receives the request for the form to be displayed.

UC 10 Downloading Form

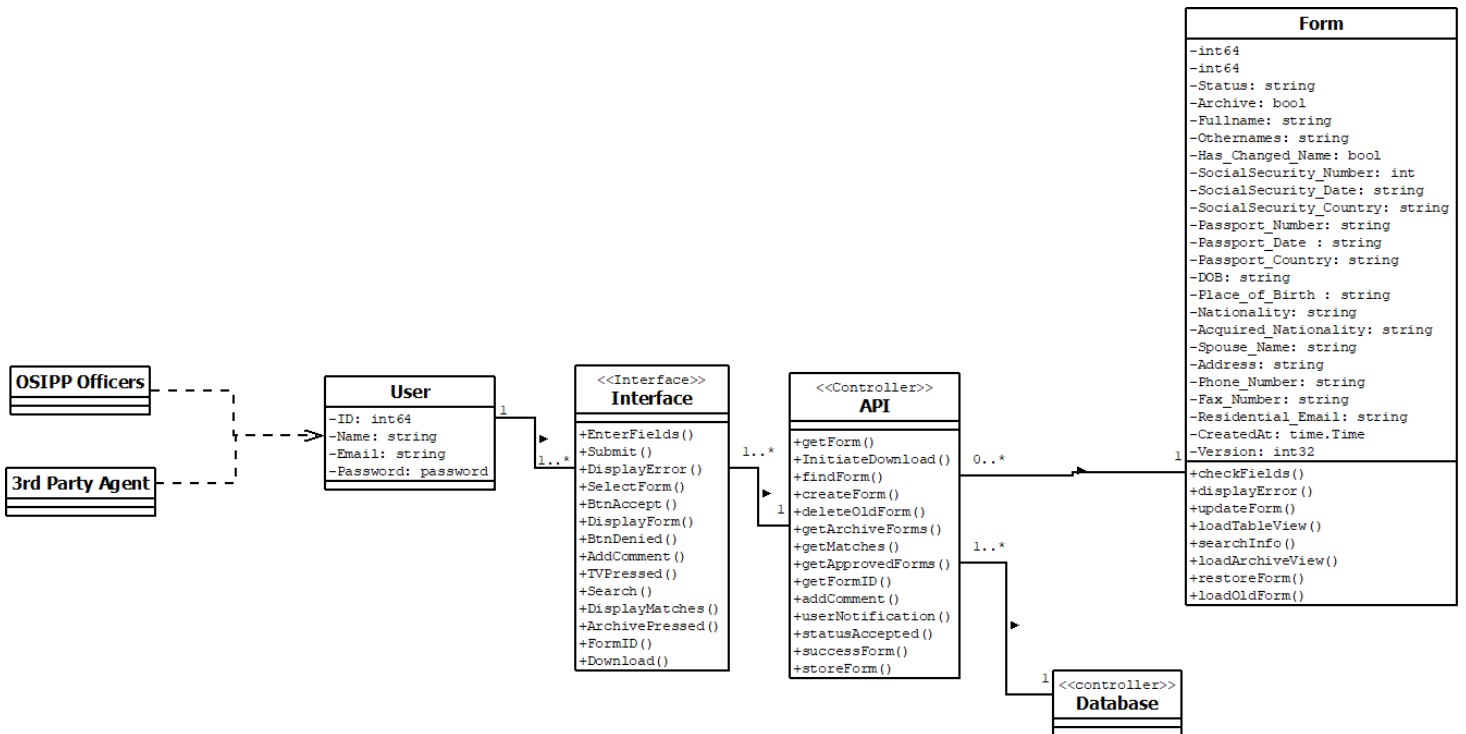


Description for UC-10:

The OSIPAdmin clicks the download button within the UI of the main page, then it sends a request to the PDFCONTROLLER to get the ID of the form from the database for the conversion to commence. The PDFCONVERTER begins the conversion from an electronic form to a downloadable version so the hard copy can be utilized.

CLASS DIAGRAMS AND INTERFACE SPECIFICATIONS

Class Diagram:



User
-ID: int64
-Name: string
-Email: string
-Password: password

<<Interface>> Interface
+EnterFields() +Submit() +DisplayError() +SelectForm() +BtnAccept() +DisplayForm() +BtnDenied() +AddComment() +TVPressed() +Search() +DisplayMatches() +ArchivePressed() +FormID() +Download()

<<Controller>> API
+getForm() +InitiateDownload() +findForm() +createForm() +deleteOldForm() +getArchiveForms() +getMatches() +getApprovedForms() +getFormID() +addComment() +userNotification() +statusAccepted() +successForm() +storeForm()

Form
-int64 -int64 -Status: string -Archive: bool -Fullname: string -Othernames: string -Has_Changed_Name: bool -SocialSecurity_Number: int -SocialSecurity_Date: string -SocialSecurity_Country: string -Passport_Number: string -Passport_Date : string -Passport_Country: string -DOB: string -Place_of_Birth : string -Nationality: string -Acquired_Nationality: string -Spouse_Name: string -Address: string -Phone_Number: string -Fax_Number: string -Residential_Email: string -CreatedAt: time.Time -Version: int32
+checkFields() +displayError() +updateForm() +loadTableView() +searchInfo() +loadArchiveView() +restoreForm() +loadOldForm()

- **UC1-FillingForm**

All fields in the form must be filled in order for the system to correctly capture all fields. The system will check all the fields, verifying none are missing before the 3rd party agent can submit. The picture below shows an illustration of the Use case.



OFFICE OF THE SUPERVISOR OF
INSURANCE & PRIVATE PENSION



User

Form ID: 2045783820

BIOGRAPHICAL AFFIDAVIT FORM

- Affiant's Full Name:
- Other names used at any time:
- Have you ever had your name changed? ☐ Yes ☒ No
If "Yes", (a) provide previous name(s) and (b) give the reason for the change.
- Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.

Document	Number of Document	Date of issuance	Country of issuance
Social Security	23262988	2/15/21	Belize
Passport	P00035684	10/28/20	Belize
National Health Insurance	N/A	N/A	N/A
Other, specified			
- Date of Birth: (DD/MM/YYYY)
- Place of Birth: include District/State and Country
- Nationality: Indicate how acquired.
☒ Birth
☐ Naturalization
☐ Marriage
☐ Other, specify
- Spouse's Name:
- Affiant's Address:

Physical Address	Telephone No.	Fax No.	Email Address
Residential	501-223-4567	N/A	john.doe@gmail.com
Business	N/A	N/A	N/A

Signature: Date:

SUBMIT

Figure 10. Shows required fields when filling form

- **UC2-AcceptingForm**

The public office of OSIPP will review the form submitted by the 3rd party agent. The public officer will click on “Accept” if the form has the correct information. The system will then move the form from unverified(Pending) to verified database.

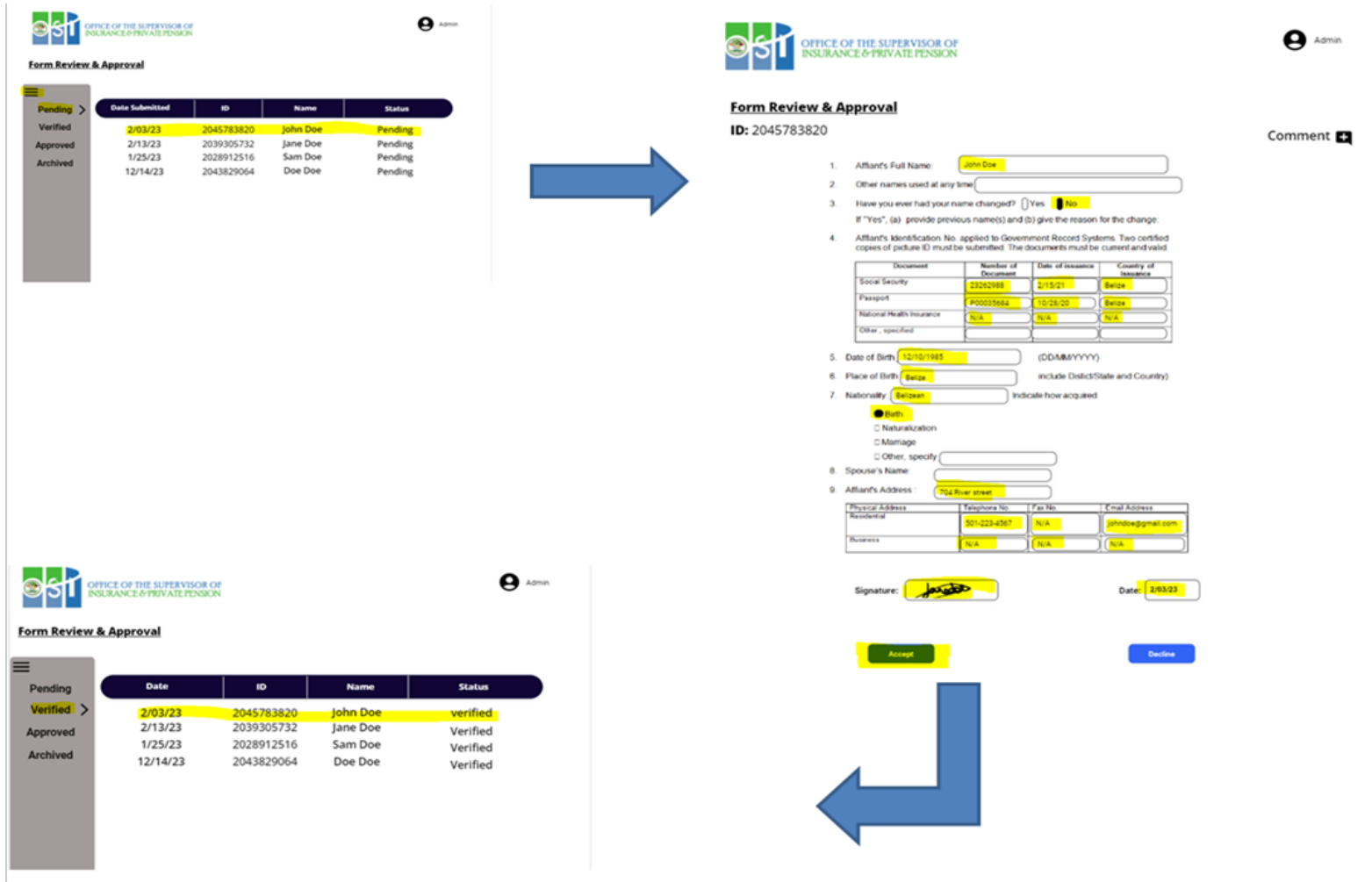



Figure 11. Shows steps taken by the public officer of OSIPP to accept a form

- **UC5- StoringData**

The public officer will click on the “Approve” button that will be available to him if the form is correctly filled out. After approval the system will display a “success” message confirming the form has been stored successfully in the database.




OFFICE OF THE SUPERVISOR OF
 INSURANCE & PRIVATE PENSION

Admin

Form Review & Approval

ID: 2045783820

Comment 


- Affiant's Full Name:
- Other names used at any time:
- Have you ever had your name changed? ☐ Yes ☒ No
If "Yes", (a) provide previous name(s) and (b) give the reason for the change:
- Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.

Document	Number of Document	Date of issuance	Country of Issuance
Social Security	2302988	2/15/21	Belize
Passport	P00035664	10/28/20	Belize
National Health Insurance	N/A	N/A	N/A
Other, specified			
- Date of Birth: (DDMMYYYY)
- Place of Birth: (include District/State and Country)
- Nationality: ☒ Birth ☐ Naturalization ☐ Marriage ☐ Other, specify:


Full successfully named

OK
- Spouse's Name:
- Affiant's Address:

Physical Address	Telephone No.	Fax No.	Email Address
Residential	501-223-4567	N/A	john.doe@gmail.com
Business	N/A	N/A	N/A

Signature:  Date:





OFFICE OF THE SUPERVISOR OF
 INSURANCE & PRIVATE PENSION

Admin

Form Review & Approval





	Date	ID	Name	Status	
Pending					
Verified					
Approved >	2/03/23	2045783820	John Doe	Approved	
Archived	2/13/23	2039305732	Jane Doe	Approved	
	1/25/23	2028912516	Sam Doe	Approved	
	12/14/23	2043829064	Doe Doe	Approved	

Figure 12. Shows steps taken by the public officer of OSIPP to approve a form

- **UC11-Making Account**

Site Administrators will be creating an OSIPP officer's email address and password in order for them to successfully access the forms with admin privileges.



Create Account

Username

Email

Password

Register

Data Types and Operation Signature

Traceability Matrix:

	User	Interface	API	OSIPP Officers	3rd party agents	Form
User Interface	X	X		X	X	X
Controller		X	X			X
Form Handler						X
Database Model			X			X

Design Patterns:

The class diagram is one of the simplest and easiest representations one can employ for software engineering. Where each entity gets broken down into three parts, its name, attributes, and methods. Using that simple structure, complex systems in the early stages of development can be designed in a way that is easily understood. Entities can relate to each other through various types of relationships, where arrows denote the ordering of the relationship. Along with numerated symbols placed on each end of the arrow relationship between entities denoting how much of the entities can be related to each other in the given relationship.

OCL Contract Specification:

Contract name	FillForm()
Cross Reference	3rd party user
Invariants	Actor is a 3rd party user
Precondition	Access Form
Post condition	Verify fields are filled

Contract name	AcceptForm()
Cross Reference	OSIPP Admin
Invariants	Form is Accepted
Precondition	Submit form
Post condition	Added to database

Contract name	DenyForm()
Cross Reference	OSIPP Admin
Invariants	Form is Denied
Precondition	Submit form
Post condition	Return form to 3rd party agent

Contract name	AddComment()
Cross Reference	OSIPP Admin
Invariants	Admin adds comment
Precondition	Submit form
Post condition	Return form to 3rd party agent

Contract name	StoreData()
Cross Reference	OSIPP Admin
Invariants	System Stores data
Precondition	Successful verification
Post condition	Add to database

Contract name	ToggleView()
Cross Reference	OSIPP Admin
Invariants	View changes
Precondition	Add to database
Post condition	Toggle view

Contract name	CancelForm()
Cross Reference	3rd party user
Invariants	Form is closed
Precondition	Open form
Post condition	Close and clear form

Contract name	Submit form()
Cross Reference	3rd party user
Invariants	System delivers form
Precondition	Fill all fields
Post condition	Verification

Contract name	ViewArchive()
Cross Reference	OSIPP Admin
Invariants	Records can be viewed
Precondition	Add to database
Post condition	View records

Contract name	ArchiveFile()
Cross Reference	OSIPP Admin
Invariants	Record can be hidden
Precondition	Add to database
Post condition	View archive

Contract name	RestoreForm()
Cross Reference	Restoration code
Invariants	Form sent for correction
Precondition	Submit form
Post condition	Edit invalid fields

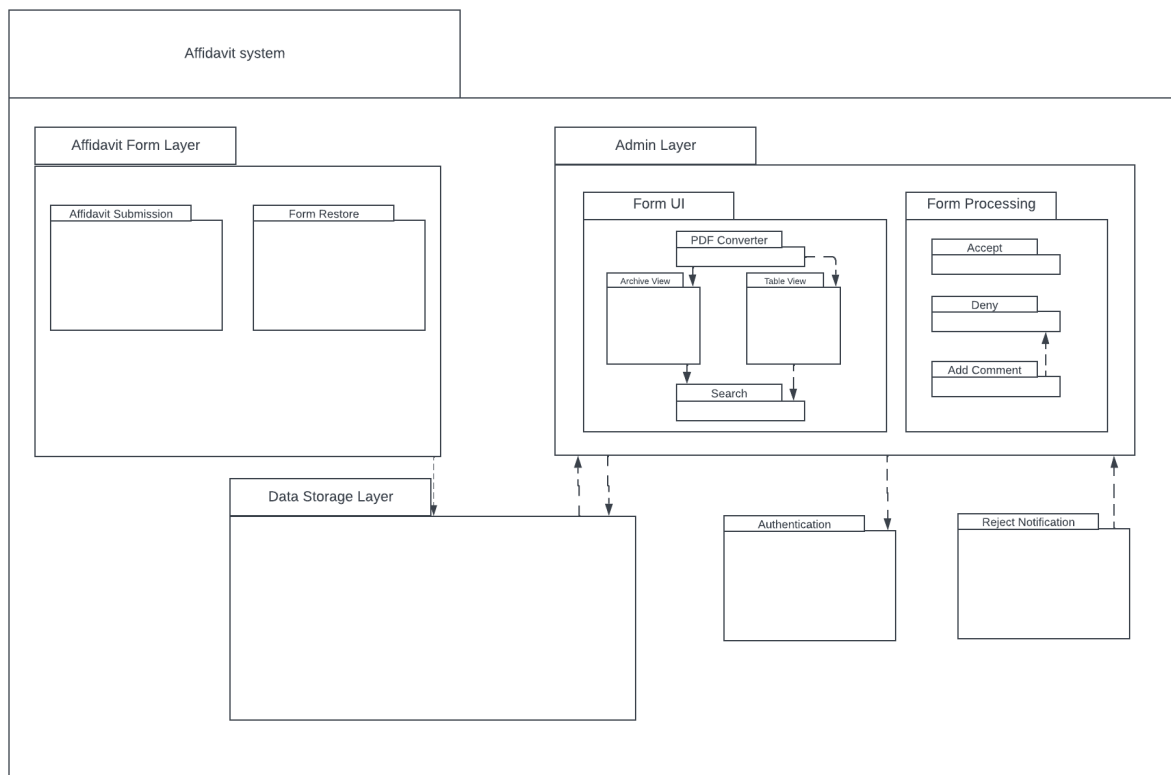
Contract name	DownloadForm()
Cross Reference	OSIPP Admin
Invariants	Form is saved to a medium
Precondition	Add to database
Post condition	View download

Contract name	MakeAccount()
Cross Reference	OSIPP Admin
Invariants	Log into application using provided credentials
Precondition	Valid email address
Post condition	Login credentials

SYSTEM ARCHITECTURE AND SYSTEM DESIGN

Identifying Subsystems & Architecture Styles:

UML Package Diagram



+

The UML Package diagram above is a high-level package view of the Affidavit system. Our system is made up of 5 Logical packages. Our first package is the Affidavit Form layer which has two subsystems. One is for Affidavit submission and one takes care of the Form Restoration. This package has one dependency, the Data storage layer because it needs to access data in order to restore the form based on its ID or to send the submitted form.

The Admin layer has two sub-packages named Form UI and Form Processing. Form UI houses our PDF converter, Archives view, Table view, and Search packages. Archives and tables both have a dependency on search. PDFconverter has dependencies on both the Archives view and Tables view. Form Processing takes care of Accepting, denying, and adding comments to the

form. Adding comments has a dependency on Deny. The Admin layer has dependencies with Data storage and with Authentication.

Data storage has a dependency on the Admin layer. Lastly, our Rejection notification package has a dependency on the Admin UI since it cannot be executed without the form being denied.

Architecture Styles:

The software architectural style employed in the design is a three-tier (Client-Server) architecture: presentation, application, and data. The presentation tier includes Admin, Market, and Vendor UI, which displays information about services like login, form data entry, and verification. The application tier includes the ability to generate reports or search 3rd party agent details, as well as view which forms are incomplete, completed, or canceled. Finally, the data tier would include local server storage of third-party agent details. REST is an abbreviation for Representational State Transfer, and API is an abbreviation for Application Program Interface. REST is a software architectural style that specifies the rules for developing web services. RESTful web services are web services that adhere to the REST architectural style. It enables requesting systems to access and manipulate web resources through the use of a consistent and predefined set of rules. REST-based systems communicate using the Internet's Hypertext Transfer Protocol (HTTP).

Mapping Subsystems to Hardware:

The system will be housed on a server at the OSIPP office, filling out the form, processing the form and accessing the information will all be done via a client. All the different clients will be serviced by the server using an API to communicate the necessary functionality. We will be using Linux based CentOS, Apache, PHP, and MySQL.

Connectors and Network Protocols:

Connectors → Physical connection CAT5 wire for data transfer

Network Protocols → include HTTP(HyperText Transfer Protocol) used to display the appropriate page and FTP(FileTransferProtocol) which allows data to be inserted into the database. Also,

Global Control Flow:

Execution Order

Our biographical affidavit form web application is, for the most part, event-driven, meaning that in order for any action to be taken by the system, an event must be detected to initiate the action. The events are mainly triggered by the users of the system; for instance, filling out a form depends entirely on the 3rd party agent's interaction with the system to submit. Most object states within the system depend entirely on event-driven updates. Before a submitted form can be reviewed, the OSIPP public office will first need an account; only then can submissions be accepted, approved, archived, and unachieved.

Time Dependency

The system we are creating is one that will have multiple people interacting with it, inputting and verifying sensitive information. Due to this we will not be having any timer on our system. 3rd party agents will be able to take as much time to fill out a form in order for mistakes not to occur, OSIPP public officers will also have plenty of time when reviewing forms to make sure all required information is available. As far as the system goes it will not have any concerns with real time other than the form submission and approval date.

Hardware Requirements:

Recommended → Hosting a web server to fill out forms requires minimal power so we have decided to run on a 2 core 2GB RAM 40GB HDD bandwidth 1Mbps

Minimum Requirements → 1 Core 1GB RAM, 20GB HDD Bandwidth 56 kbps

ALGORITHMS AND DATA STRUCTURES

Architectural Styles:

The system was designed to follow the REST(Representational State Transfer) Architecture, it was chosen because adapting our design to the architecture wouldn't be changeling. The architecture mainly revolves around interactions between the system's API and its front and back ends. Since the system will facilitate the entering and managing of information, this architecture is the best choice.

Data Structures:

Since the system uses a RESTful approach, there aren't that many complex data structures. The main data types of the system are simple abstract data types. These abstractions are representations of the data that will be stored in the database of the system. Hash Tables are utilized for the password systems of the system, followed by Key maps for error logging and general logging.

Data structures utilized:

- Abstract Data Type
- Hash Table
- Key Map

USER INTERFACE DESIGN AND IMPLEMENTATION

Modifications:

Initially, the affidavit form and Review and Approval interface were designed using a UI designer. However, the current design has been implemented using HTML/CSS. As part of the implementation process, We made significant improvements to the form's usability by adding placeholders to the insertion boxes, making it less likely for users to make mistakes when filling out the form. Furthermore, we made slight changes to the Review and Approval interface by incorporating a search bar to enhance the user experience and enable quicker access to specific forms.

GUI Designs:

Fill form

- The Diagram below displays the Biographical Affidavit form

Biographical Affidavit

1. Affiant's Full Name:	<input type="text" value="Full name"/>																				
2. Other names used at any time:	<input type="text" value="Other name"/>																				
3. Have you ever had your name changed?	<input type="radio"/> Yes <input type="radio"/> No																				
If "Yes", please provide the following information:																					
Previous name(s):	<input type="text"/>																				
Reason for the change:	<input type="text"/>																				
4. Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.																					
<table><thead><tr><th>Document</th><th>Number of Document</th><th>Date of issuance</th><th>Country of issuance</th></tr></thead><tbody><tr><td>Social Security</td><td>23262988</td><td>2/15/21</td><td>Belize</td></tr><tr><td>Passport</td><td>P00035684</td><td>10/28/20</td><td>Belize</td></tr><tr><td>National Health Insurance</td><td>N/A</td><td>N/A</td><td>N/A</td></tr><tr><td>Other ,specify</td><td></td><td></td><td></td></tr></tbody></table>	Document	Number of Document	Date of issuance	Country of issuance	Social Security	23262988	2/15/21	Belize	Passport	P00035684	10/28/20	Belize	National Health Insurance	N/A	N/A	N/A	Other ,specify				
Document	Number of Document	Date of issuance	Country of issuance																		
Social Security	23262988	2/15/21	Belize																		
Passport	P00035684	10/28/20	Belize																		
National Health Insurance	N/A	N/A	N/A																		
Other ,specify																					
5. Date of Birth:	<input type="text" value="DD/MM/YYYY"/>																				
6. Place of Birth:	<input type="text" value="District/State and Country"/>																				
7. Nationality: (Indicate how aquired.)	<input type="text" value="Birth"/> <input type="button" value="v"/>																				
8. Spouse's Name:	<input type="text" value="Name"/>																				
9. Affiant's Address:																					
<table><thead><tr><th>Physical Address</th><th>Telephone Number</th><th>Fax Number</th><th>Email Address</th></tr></thead><tbody><tr><td>Physical Address</td><td>501-202-4532</td><td>N/A</td><td>helen.g@btl</td></tr></tbody></table>	Physical Address	Telephone Number	Fax Number	Email Address	Physical Address	501-202-4532	N/A	helen.g@btl													
Physical Address	Telephone Number	Fax Number	Email Address																		
Physical Address	501-202-4532	N/A	helen.g@btl																		

View Unverified

- The Diagram below displays the Unverified interface.

Unverified

Verified

Archive

Search

Unverified

ID	FNAME	LNAME	DATE	STATUS
1	John	Doe	2022-01-01	Pending
2	Jane	Smith	2022-02-01	Pending
3	Bob	Johnson	2022-03-01	Pending
4	Alice	Williams	2022-04-01	Pending

View Verified

- The Diagram below displays the verified interface.

Unverified

Verified

Archive

Verified

ID	FNAME	LNAME	DATE	STATUS
1	John	Doe	2022-01-01	Verified
2	Jane	Smith	2022-02-01	Verified
3	Bob	Johnson	2022-03-01	Verified
4	Alice	Williams	2022-04-01	Verified

View Archive

- The Diagram below displays the Archive interface.

Unverified

Verified

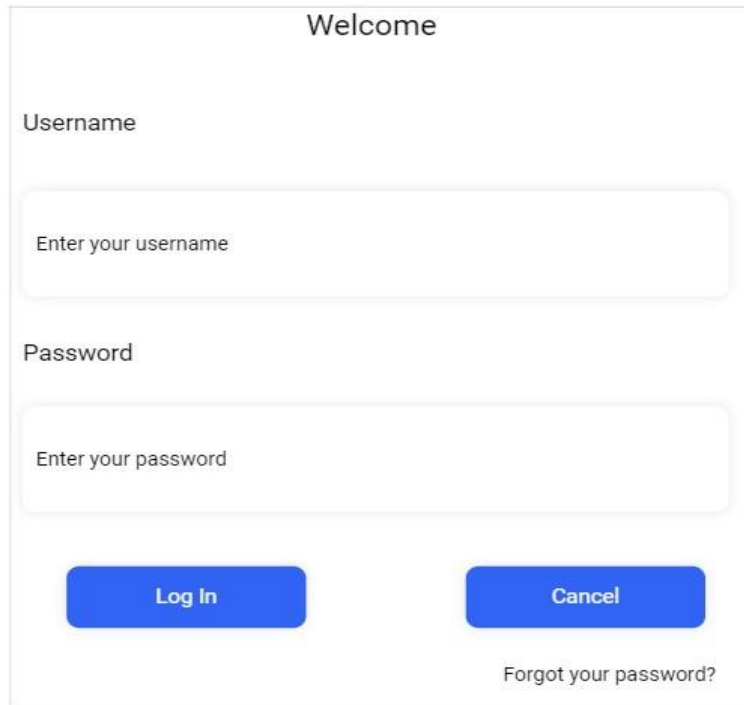
Archive

Archive

ID	FNAME	LNAME	DATE	STATUS
1	John	Doe	2022-01-01	Verified

Login

- The Diagram below displays the login screen for OSSIP Officers to access system



The diagram shows a login interface with a light gray background. At the top, the word "Welcome" is centered. Below it, the label "Username" is followed by a white input field with a light gray border and the placeholder text "Enter your username". Below this, the label "Password" is followed by a similar white input field with the placeholder text "Enter your password". At the bottom, there are two blue buttons with white text: "Log In" on the left and "Cancel" on the right. Below the "Cancel" button, the text "Forgot your password?" is displayed.

Welcome

Username

Enter your username

Password

Enter your password

Log In

Cancel

Forgot your password?

Restore Form

- The Diagram below displays a form being restored after selecting an Affiant name.

Unverified

Verified

Archive

Search

Unverified

ID	FNAME	LNAME	DATE	STATUS
1	John	Doe	2022-01-01	Pending
2	Jane	Smith	2022-02-01	Pending
3	Bob	Johnson	2022-03-01	Pending
4	Alice	Williams	2022-04-01	Pending

1. Affiant's Full Name: John doe

2. Other names used at any time: N/A

3. Have you ever had your name changed? ☐ Yes ☒ No

If "Yes", please provide the following information:

Previous name(s): N/A

Reason for the change: N/A

4. Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.

Document	Number of Document	Date of issuance	Country of issuance
Social Security	23262988	2/15/21	Belize
Passport	P00035684	10/28/20	Belize
National Health Insurance	N/A	N/A	N/A
Other ,specify			

5. Date of Birth: 12/10/1985

6. Place of Birth: Belize

7. Nationality: (Indicate how aquired.) Birth

8. Spouse's Name: Name

9. Affiant's Address:

Physical Address	Telephone Number	Fax Number	Email Address
Residential	501-202-4532	N/A	johndoe@gmail.com
Business	N/A	N/A	N/A

Add comment

- The Diagram below displays a comment being added to the form because the affiant missed some fields.

1. Affiant's Full Name:

2. Other names used at any time:

3. Have you ever had your name changed? ☐ Yes ☒ No
If "Yes", (a) provide previous name(s) and (b) give the reason for the change.

4. Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.

Document	Number of Document	Date of issuance	Country of issuance
Social Security	<input type="text" value="23262988"/>	<input type="text" value="2/15/21"/>	<input type="text" value="Belize"/>
Passport	<input type="text" value="P00035684"/>	<input type="text" value="10/28/20"/>	<input type="text" value="Belize"/>
National Health Insurance	<input type="text" value="N/A"/>	<input type="text" value="N/A"/>	<input type="text" value="N/A"/>
Other, specified	<input type="text"/>	<input type="text"/>	<input type="text"/>

5. Date of Birth: (DD/MM/YYYY)

6. Place of Birth: include District/State and Country)

7. Nationality: Indicate how acquired.
☒ Birth
☐ Naturalization
☐ Marriage
☐ Other, specify:

8. Spouse's Name:

9. Affiant's Address:

Physical Address	Telephone No.	Fax No.	Email Address
Residential	<input type="text" value="501-223-4567"/>	<input type="text" value="N/A"/>	<input type="text" value="johndoe@gmail.com"/>
Business	<input type="text" value="N/A"/>	<input type="text" value="N/A"/>	<input type="text" value="N/A"/>

Signature:

Date:

Comment

Form Has missing fields. Please add information and resubmit

Deny and Accept From

- The Diagram below displays a form that is being reviewed by the OSIPP officer.

1. Affiant's Full Name:

2. Other names used at any time:

3. Have you ever had your name changed? ☐ Yes ☒ No
If "Yes", (a) provide previous name(s) and (b) give the reason for the change:

4. Affiant's Identification No. applied to Government Record Systems. Two certified copies of picture ID must be submitted. The documents must be current and valid.

Document	Number of Document	Date of issuance	Country of issuance
Social Security	23262988	2/15/21	<input type="text" value="Belize"/>
Passport	P00035684	10/28/20	<input type="text" value="Belize"/>
National Health Insurance	N/A	N/A	N/A
Other, specified	<input type="text"/>	<input type="text"/>	<input type="text"/>

5. Date of Birth: (DD/MM/YYYY)

6. Place of Birth: (include District/State and Country)

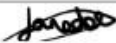
7. Nationality: (Indicate how acquired)

☒ Birth
☐ Naturalization
☐ Marriage
☐ Other, specify:

8. Spouse's Name:

9. Affiant's Address:

Physical Address	Telephone No.	Fax No.	Email Address
Residential	501-223-4567	N/A	johndoe@gmail.com
Business	N/A	N/A	N/A

Signature: 

Date:

Ease of Use:

To ensure ease of use, the affidavit system will feature a user-friendly interface accessible to all users. The platform will be adaptable to different interfaces, ensuring excellent usability for everyone. In designing the OSIPP officer dashboard UI, we will prioritize simplicity and clarity to facilitate quick and easy learning for users. By prioritizing clear and intuitive design, we aim to create a system that is efficient, effective, and user-friendly for all.

Test Case Design**Plan for testing Algorithm, Non-Functional and User Interface Requirements**

DESIGN OF TESTS

This design test stage will employ fourteen use cases. The tested functions will provide critical feedback on how each function will respond when the user interacts with the system. The test case will refer to one or more use cases. Other cases will be used as part of the testing procedures in fourteen cases. Each test case will pass or fail based on a fail or success scenario. As a result, the design test intends to provide the scope of the system's implementation.

Test Case 1

Test Case	TC-1
Use Case in Test	UC-1: FillingForm
Criteria for success/fail	Test is successful once the 3rd party agent fills all fields of the form
Input Data:	User Info
Test Procedure:	Expected Output
Step 1: Leaving "Affidavit's Fullname" field blank	Display error message "Fullname field can't be blank"
Step 2: Leaving half of the form filled out	Display error message "kindly verify that all information is entered"
Step 3: Filling all fields of the form to be submitted	Display message "Form was submitted"
This test case tests the functionality of verifying each field of the form that was filled(UC-1).	

Test Case 2

Test Case	TC-2
Use Case in Test	UC-2: AcceptingForm
Criteria for success/fail	Test is successful once the 3rd party agent has all the necessary fields filled in the form
Input Data:	Accept Button
Test Procedure	Expected Output

Step 1: Accepting a blank form	Display error message “unable to accept a blank form”
Step 2: Accepting a form that all necessary fields are filled	Display Message “Form Accepted”
Step 3: Accepting a form that was incomplete	Display error message “unable to accept an incomplete form”
This test case tests the functionality of the accept button once the user pressed it(UC-2).	

Test Case 3

Test Case	TC-3
Use Case in Test	UC-3: DenyingForm
Criteria for success/fail	Test is successful once the form can denied after once information entered was incorrect
Input Data:	Return Button
Test Procedure	Expected Output
Step 1: Denying the form if the data is inaccurate	Display “Form has been returned”
This test case tests the web application ability to deny the form properly that is returned back to the 3rd party agent(UC-3).	

Test Case 4

Test Case	TC-4
Use Case in Test	UC-4: AddingComments
Criteria for success/fail	Test is successful once the public officer can add comments to each field of the form during the verification phase
Input Data:	Insert Comment Option
Test Procedure	Expected Output
Step 1: Adding comments to notify the user that “Date of Birth” is incorrect	Display comment beside “date of birth” field

Step 2: Adding a blank comment	No comment gets added
Step 3: Edit comments	Popup Menu displayed “edit comment”
This test case tests the web application’s ability to add comments(UC-4) to any field of the form.	

Test Case 5

Test Case	TC-5
Use Case in Test	UC-5: StoringData
Criteria for success/fail	Test that the application can store the data to the database after it passed the verification phase
Input Data:	None
Test Procedure	Expected Output
Step 1: Attempt to store data if the database is offline	Display message “unable to reach data server”
Step 2: Attempts to store data if the database is active	Display message to user “form has been stored”
This test case tests the web application’s ability to store the information into the database(UC-5).	

Test Case 6

Test Case 6	TC-6
Use Case in Test	UC-6: TogglingTableView
Criteria for success/fail	Test that the web application can display all forms that were approved by the public officer
Input Data	ToggleFormView Button
Test Procedure	Expected Output

Step 1: Selecting the option to display all approved forms	Another window will open to display all approved forms
Step 2: Attempting to view approved forms if the server is down	Display error message “Unable to view approved forms”
This test case tests the web application ability to view all approved forms in a readable manner(UC-6).	

Test Case 7

Test Case 7	TC-7
Use Case in Test	UC-7: CancelingForm
Criteria for success/fail	Test that the function can not only cancel the form but to also clear all fields.
Input Data	Click “Cancel Form”
Test Procedure	Expected Output
Step 1: Pressing Cancel when the form is completely filled	Display message “Form has been cancel” all fields are cleared
Step 2: Pressing Cancel when the form is already empty	Display message “No field(s) have been filled” void cancelation
This test case tests the cancelForm function that not only checks if the form is already filled but also ensures that all fields are cleared(UC-7).	

Test Case 8

Test Case 8	TC-8
Use Case in Test	UC-8: Submitting form
Criteria for success/fail	Test that the function can submit the form once the 3rd party agent fills it out properly
Input Data	Press Submit button
Test Procedure	Expected Output

Step 1: Attempts to submit a form that is incomplete	Display error message “Unable to submit form, kindly ensure all fields are completed” function SubmitForm did not execute
Step 2: Attempts to submit a form that has been filled out properly	Display message “Form has been submitted” function SubmitForm executes
Step 3: Attempts to submit a form that is blank	Display error message “Unable to submit a blank form” function SubmitForm did not execute
This test case tests the function’s ability to verify that the user has filled out the form properly before it can be submitted(UC-8)	

Test Case 9

Test Case 9	TC-9
Use Case in Test	UC-9: ViewArchive
Criteria for success/fail	Test that the function can allow the user to view all archived forms in the system
Input Data	ViewArchive button
Test Procedure	Expected Output
Step 1: Attempts to view archived forms while the server is offline	Display error message “unable to view archived forms” function ViewArchive didn’t execute
Step 2: Attempts to view archived forms while the server is online	A new window pops up to view all archived forms
This test case tests the functionality of viewing all archived forms that is stored in the web application(UC-9).	

Test Case 10

Test Case 10	TC-10
Use Case in Test	UC-10: ArchivingFile
Criteria for success/fail	Test that the web application is able to

	Archive a form during its verification phase
Input Data	ArchiveForm button
Test Procedure	Expected Output
Step 1: Archiving a form while the server is offline	Display error message “unable to archive form” function ArchiveForm did not execute
Step 2: Archive a form while the server is online	Display message “Form archived successfully” function
This test case tests the functionality of Archiving a form while its being verified(UC-10)	

Test Case 11

Test Case 11	TC-11
Use Case in Test	UC-11: RestoringForm
Criteria for success/fail	Test that the function is able to restore back the form once the 3rd party user receives a unique restoration code via email
Input Data	RestoreForm button
Test Procedure	Expected Output
Step 1: User attempts to receive restoration code during power outage	Display message “unable to generate restoration code” function RestoreForm didn’t execute
Step 2: User attempts to receive restoration code	Web application sends restoration code to user’s email, function RestoreForm executes
This test case tests the functionality of sending 3rd party agents a unique restoration code to retrieve back a form(UC-11).	

Test Case 12

Test Case 12	TC-12
--------------	-------

Use Case in Test	UC-12: DownloadForm
Criteria for success/fail	Test is successful once the form can be downloaded into a PDF format for the public officer
Input Data	DownloadAffidavit button
Test Procedure	Expected Output
<p>Step 1: OSIPP public officer attempts to download the form that was filled</p> <p>Step 2: User attempts to download the form during a black out</p>	<p>Web application downloads the pdf version of the form with all user's info filled out, function DownloadAffidavit executed</p> <p>Display error message "unable to download form, please try again later", function DownloadAffidavit did not execute</p>
This test case tests the web application's functionality of converting the automated form into a downloadable pdf file for the user(UC-12).	

Test Case 13

Test Case 13	TC-13
Use Case in Test	UC-13: MakingAccount
Criteria for success/fail	Test is successful once the OSIPP public officers has already gotten a working government email address
Input Data	OSIPP government email address
Test Procedure	Expected Output
<p>Step 1: Admin attempts to create OSIPP user account with a false email address</p> <p>Step 2: Admin attempts to create OSIPP user account with an actual government email address</p>	<p>Display error message "gobmail doesn't exist"</p> <p>Display message "User account as been created" function CreateAccount was executed</p>
This test case tests the functionality of created accounts for OSIPP public officers to use the web application(UC-13).	

HISTORY OF WORK

This project is a testament to the will that students have toward completing a given assignment. The group consists of students with varying workloads inside or outside of academic life. Like a battlefield, the recruits were prepared and ready to begin the challenge of creating this system at the beginning of the semester. Pushing themselves where possible whilst also getting lost in the fog of war. A lot of time was allotted for this project to be completed, even acknowledged by the lecturer that it wouldn't be enough and he was right. The main issue doesn't stem from the students' faulting will but their endurance. Being bogged down by multiple courses, life events, and simply unfortunate circumstances outside of their control. Yet, amongst the cracks, this system was put together.

As previously mentioned all the tasks were divided from the get-go, where certain members would be in charge of specific aspects of the project. Which turned out to be a fault in our plans. Since each member had a role to play, having each member fall into the trap that is the time crunch of multiple assignments next to each other. Parts of this project suffered heavily from neglect.

In spite of this, all members had the will to press forward with completing the assignment regardless of the outcome, and the result of work. All members of the group worked well with each other, participating when needed and maintaining proper communication ensuring that work was done.

Current Status:

The system is usable and can be deployed at the moment, however, some changes might want to be made after deployment. It's able to carry out all of the functions that are required of it at this time.

Future Work:

After deployment, additions to the database will need to be made, since the biographical affidavit has been changed. The database will be needed to be altered to store more information. Since the system is a digital version of the actual biographical affidavit, not much will need to be changed unless the biographical affidavit application process is changed. Should changes be made the system is designed to be modified after deployment.

PROJECT MANAGEMENT

Overall Management of Project:

Work was divided among the group voluntarily for the most part. Other tasks were simply done by whichever group member wasn't occupied at the time. Therefore most of the tasks were usually done individually so that there would be no discrepancy in each part. Though they were done individually the same individual group member would often do related tasks ensuring that no discrepancies were made.

Given the amount of time allotted for the project, an estimate of 60% utilization of that time was used. A preferable 100% utilization would have been used however, not all members were able to work on their assigned tasks in suitable intervals. Leading to an unsteady development cycle.

All group members participated in the programming of the system, some focused on the frontend, while others focused on the backend of the system. Most of the development was done independently with the use of github to distribute parts of the project. During the final tests, the final versions of the system were distributed using zip files containing the entire code base of the program.

REFERENCES

Fielding, R. T. (2023, February 20). *Wikipedia*. Retrieved from Representational state transfer: https://en.wikipedia.org/wiki/Representational_state_transfer

Government of Belize. (2023, February 22). *Office of the Supervisor of Insurance and Private Pension*. Retrieved from Office of the Supervisor of Insurance and Private Pension: <https://osipp.gov.bz>

Geeks for Geeks. (2023, January 5). *REST API Architectural Constraints*. Retrieved from geeks for geeks: <https://www.geeksforgeeks.org/rest-api-architectural-constraints/>

Ministry of Finance. (2023, February 24). *Ministry of Finance*. Retrieved from Government of Belize: <https://www.mof.gov.bz>

Summary of Changes

Changes made from Report 1 and 2

List of key revisions made from the past two reports:

- Removed 'add comment' feature from Biographical Affidavit from user documentation
- Removed 'add comment' feature from use case and test case
- Re-added comment feature

Fixed Use Case Diagram

Added association diagram

Updated State Sequence Diagram

Updated Test Cases

Update Interaction Diagram