DanRamFar Insurance

**The Insurance Plan**

Analysis Design Document

Name(s): Daniel Walker, Ramiz Cohadarevic, Farooq Abrahem

Lab Section:

Workstation:

Date: (03/11/2015)

**TABLE OF CONTENTS**

**1. INTRODUCTION 2**

1.1 Purpose 2

1.2 Scope 2

1.3 Overview 2

1.4 Reference Material 2

1.5 Definitions and Acronyms 2

**2. SYSTEM OVERVIEW 2**

**3. SYSTEM ARCHITECTURE 2**

3.1 Architecture Design 2

3.2 Decomposition Description 2

3.3 Design Rationale 3

**4. DATA DESIGN 3**

4.1 Data Description 3

4.2 Data Dictionary 3

**5. COMPONENT DESIGN 3**

**6. HUMAN INFERFACE DESIGN 4**

6.1 Overview of User Interface 4

6.2 Screen Images 4

6.3 Screen Objects and Actions 4

**7. REQUIREMENTS MATRIX 4**

**8. APPENDICES 4**

**1. INTRODUCTION**

**1.1 Purpose**

The purpose of this SDD is to enrich knowledge to perspective clients about the functions and system designs that the program will feature. These will help instill an image explaining how the architecture of the software will work.

**1.2 Scope**

This software intends to make it easy for customers to find and purchase an insurance plan that works for them. This software also intends to give customer complete control over their account to manage such plans. The design of the software is intended make a simple user interface that is easy understood by the average user. It is also designed to give extra support to customers that need it.

**1.3 Overview**

This document intend to give an overview of how the DanRamFar software with be built and how it is intended to run. There several diagrams to help illustrate how the different modules of the software are composed and how their inner workings of the components are used.

**1.4 Reference Material**

To be added…

**1.5 Definitions and Acronyms**

To be added…

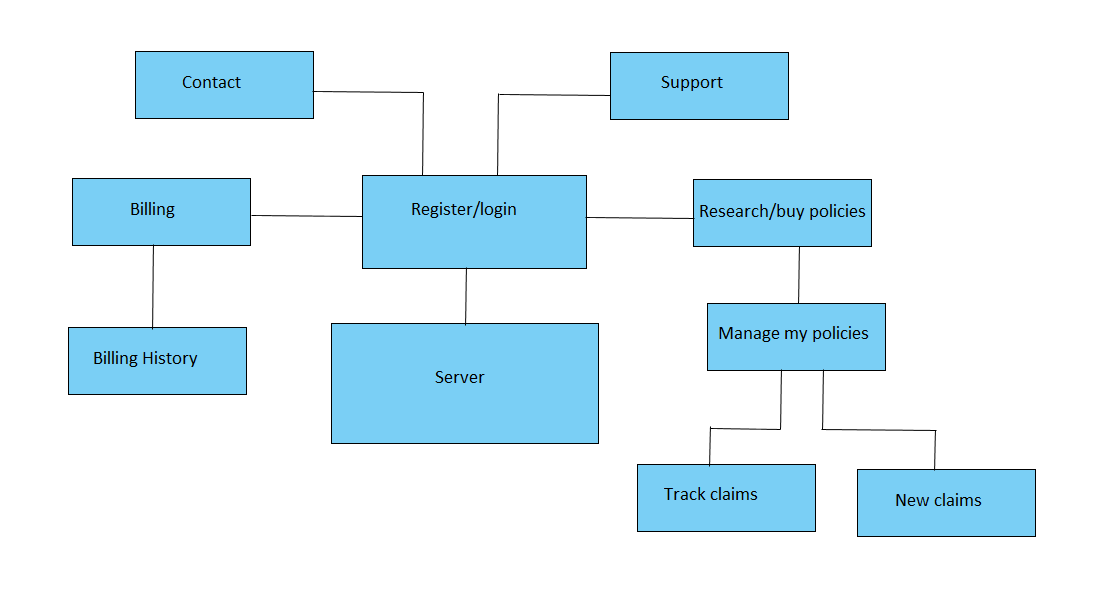
**2. SYSTEM OVERVIEW**

This software will assist customers in being able to select an insurance plan using easy to read and descriptive information. Using a simple navigation system the user will be able to navigate quickly throughout to site to important areas that relate to their account. One such feature is billing where they can pay bills. Another feature would a support page to get quick help.

**3. SYSTEM ARCHITECTURE**

**3.1 Architectural Design**

Figure below shows the general architectural design of our insurance management site. The user will be able to register or login, this module is directly connected to the system. Upon being logged in can get support or can contact an admin. They can also access billing while in billing can access the billing history. While logged in they can research and buy new policies, go further to manage those polices and new claims as well as track those claims.



**3.2 Decomposition Description**

To be added…

**3.3 Design Rationale**

To be added…

**4. DATA DESIGN**

**4.1 Data Description**

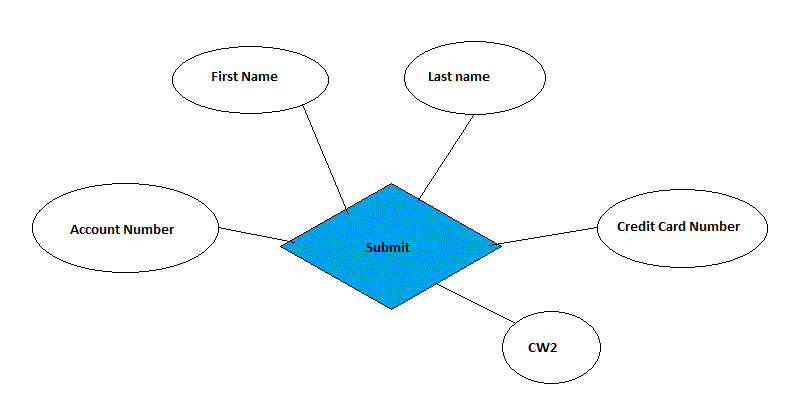
**Research / Buy Policy Module**

To be added. Each use case 1 diagram. 3 total

**Billing Module**

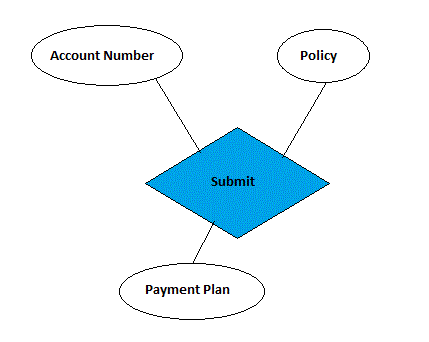
**Submit a Payment**

The following is the data design for the “Submit a Payment” use case for the billing module. The first name, last name, Account Number, Credit Card Number and CW2 are characteristics of submitting a payment.



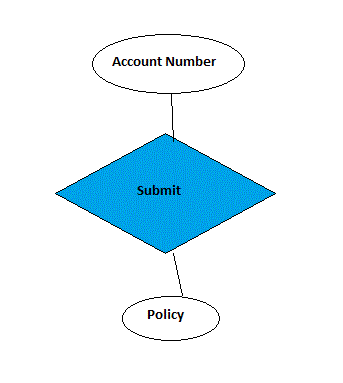
**Set a Payment Plan**

The following is the data design for the “Set a Payment Plan” use case for the billing module. The account number, policy, and payment plan are characteristics of set a payment plan.

****

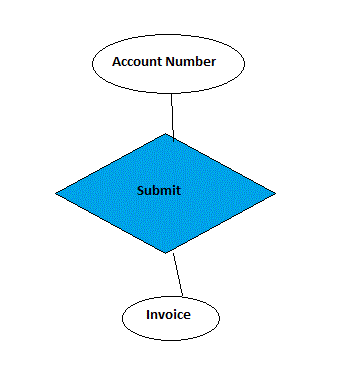
**View all Invoices**

The following is the data design for the “View all Invoices” use case for the billing module. The account number and policy are characteristics of view all invoices.



**Update Invoice**

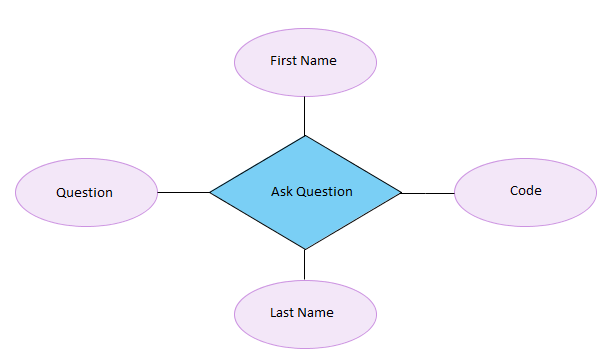
The following is the data design for the “Update Invoice” use case for the billing module. The account number and Invoice are characteristics of update invoices.

****

**Contact Module**

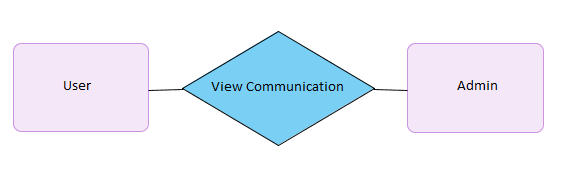
**Ask a Question**

The following is the data design for the “Ask a Question” use case for the contact module, the user asks a question is the relationship verb. The first name, last name, code, and question are characteristics of ask a question.

****

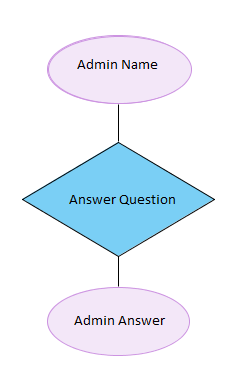
**View previous Communication**

The following is the data design for the view communication use case of the contact module. There is no input required in this use case, thus there is no attributes. The two entities user and admin share the relationship to view pervious communication made by the two, the system will simply retrieve and display requested information.

****

**Answer Question**

The following is the data design for answer question use case for the contact module, the answer question has two attributes in the admin answer and admin name. This the only information required thus the data is simple.

****

**4.2 Data Dictionary**

The following tables are defined for the Research/Buy Module

|  |  |  |  |
| --- | --- | --- | --- |
| **User Table** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **Cash Recovery** | double | Not null | The limit of cash recovery |
| **Medical Stuff** | double | Not null | The limit of medical fees |

|  |  |  |  |
| --- | --- | --- | --- |
| **User/Admin Table** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **access\_id** | INT | Not Null | Unique identifier of a user.either an Admin or user |

The following tables are defined for the Billing Module

|  |  |  |  |
| --- | --- | --- | --- |
| **User** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **Account\_Num** | INT (10) | Not null | PK. Account Number |
| **First\_name** | varchar(35) | Not null | User’s First Name |
| **Last\_Name** | varchar(35) | Not null | User’s Last Name |

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **Admin\_id** | INT | Not null | PK. Admin ID |

|  |  |  |  |
| --- | --- | --- | --- |
| **Account** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **Account\_Num** | INT (10) | Not null | PK. Account Number |
| **Policy\_ID** | INT (20) | Not null | ID of Insurance Policy |
| **Credit\_CardNUM** | String (16) | Not null | Credit Card Number |
| **CW2\_Num** | INT (3) | Not null | CW2 Security Number |
| **Payment\_Plan** | String | Not null | Payment Plan |

|  |  |  |  |
| --- | --- | --- | --- |
| **Policy** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **Policy\_ID** | INT (20) | Not null | PK. ID of Insurance Policy |
| **Policy\_Type** | String | Not null | Type of Policy |
| **Invoice\_Num** | Int (20) | Not null | Invoice Number |
| **Invoice\_Data** | varchar(500) | Not null | Invoice Data |

The following tables are defined for the Contact Module

|  |  |  |  |
| --- | --- | --- | --- |
| **User** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **User\_id** | INT | Not null | PK. Unique identifier of a user. |
| **First\_name** | varchar(25) | Not null | User’s first name |
| **Last\_Name** | varchar(35) | Not null | User’s last name |
| **Question** | varchar(100) | Not null | User’s question |
| **Code** | varchar(9) | Not null | Code to be typed by user to reduce spam |

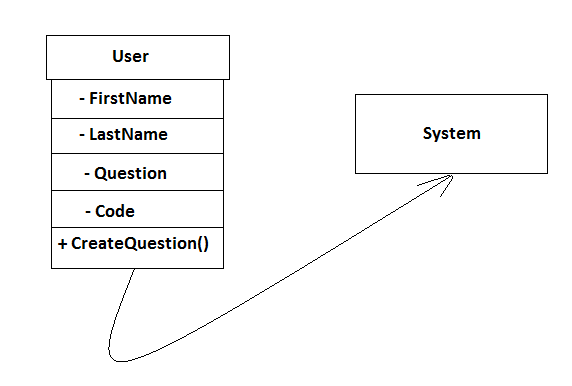
|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** |  |  |  |
| **Entity** | **Type** | **Value** | **Description** |
| **Admin\_id** | INT | Not null | PK. Unique identifier of an admin. |
| **Admin\_name** | varchar(20) | Not null | Admin’s name |
| **Admin\_answer** | varchar(200) | Not null | Admin’s answer |

Insert Full entity relationship for ALL MODULES assigned to group.

**5. COMPONENT DESIGN**

**Contact Module**

**Ask a question (user)**

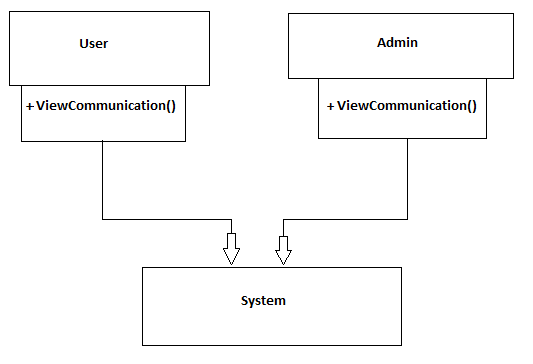
****

**Entities:**

**User-** User represents only the potential customers for the insurance offered. This use case has the job of creating questions customer may ask.

**System-** System validates input has been entered correctly, if done so question will be created, otherwise error message will be returned.

**View previous communication (both)**

****

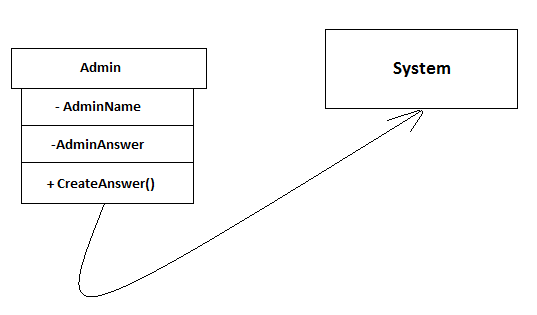
**Entities:**

**User-** User represents only the customers currently registered and logged in on the insurance site. The job of this use case is to allow the user to view all previous communication, no input is necessary.

**Admin-** Admin represents only the admins currently logged in on the insurance site. The job of this use is to allow the admin to view all previous communication.

**System-** System requires no validation and will retrieve and display all previous communication made by these two entities.

**Answer Questions (admin)**

****

**Entities:**

**Admin-** Admin represents only the people in charge of the insurance company who have the power to answer customer questions. The job of this use case is to create an answer to any question asked by the user.

**System-** System validates input requirements are met and creates an answer to a question.

**6. HUMAN INTERFACE DESIGN**

**6.1 Overview of User Interface**

**Research / Buy Policy Module**

Info about Module describing functions.

**Billing Module**

These use cases were designed so that the user can easily put in there information, and the system will then assist them in finishing the process of paying for their invoices. We choose this style of formatting because it is simple and does not require a lot of input.

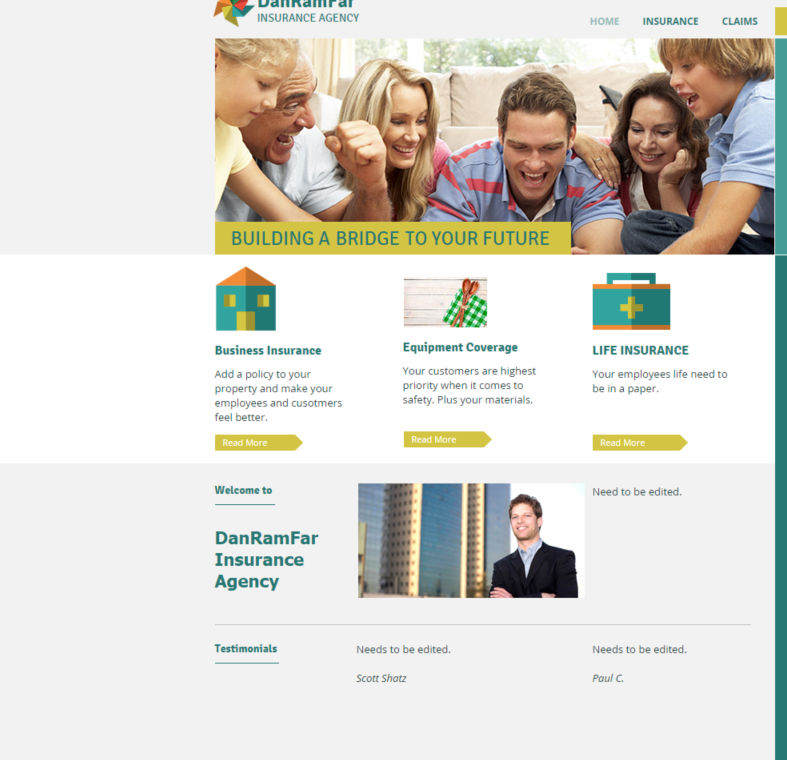
**Contact Module**

The use case design was done in this manner because I wanted to follow one of the rules of keeping it simple but not too simple. It was not necessary to make it more complex because the use cases themselves are very straight forward. There is sufficient input requested for the appropriate use cases so the user will have no problem with submitting a question or viewing any they might have asked previously, the admin likewise will have no problems in answering any question or viewing any previously answered questions.

**6.2 Screen Images**

**Research / Buy Policy Module**

**Use case – Read More (user)**



**Use case – Configure(user)**

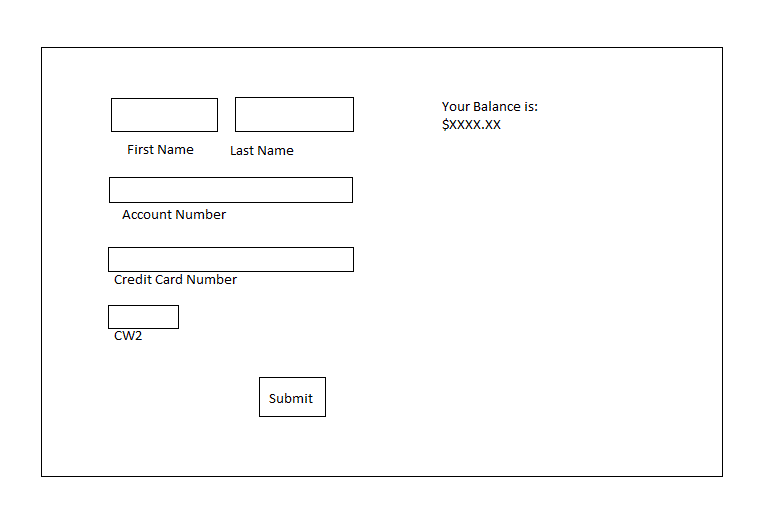


**Use case – Buy Policy (user)**

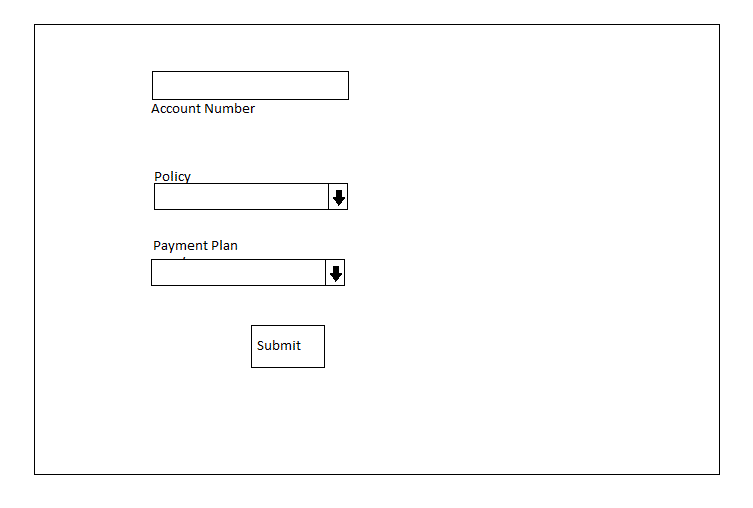
To be added…

**Billing Module**

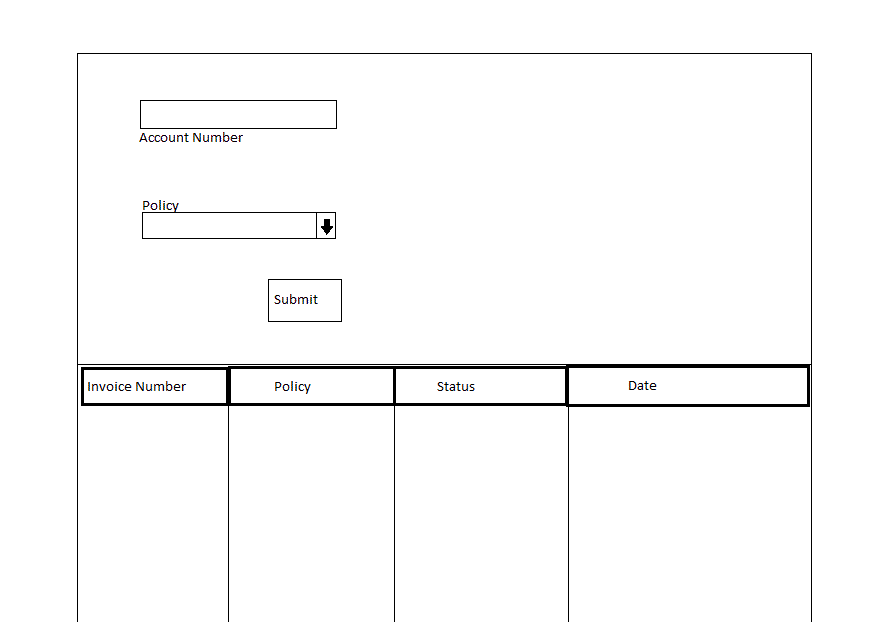
**Use case – Submit a Payment (user)**

****

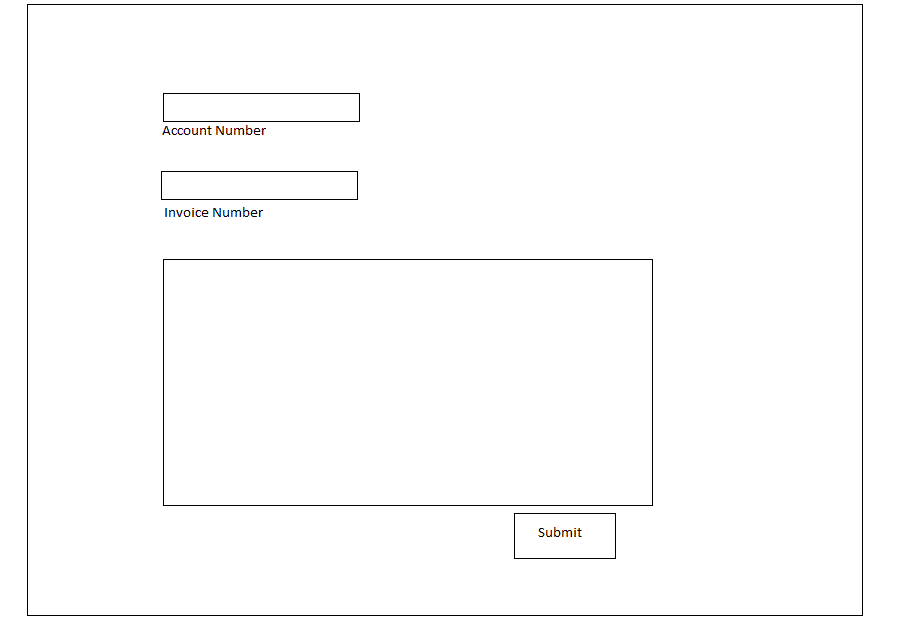
**Use case – Set a Payment Plan (user)**

****

**Use case – View Invoice Policies (user)**

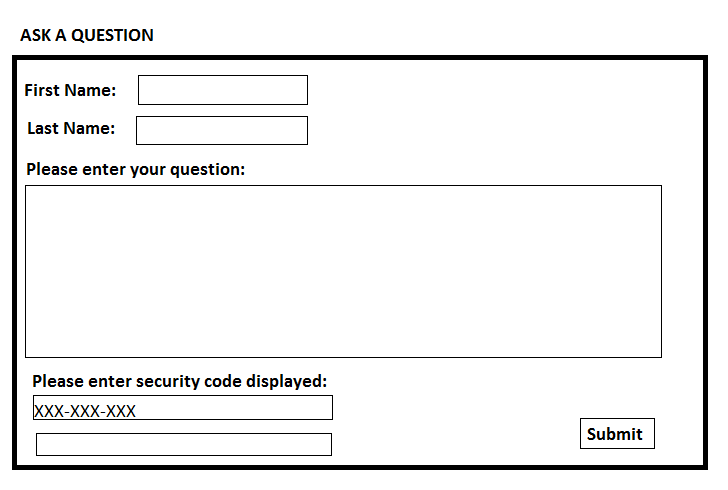
****

**Use case – Update Invoice (admin)**

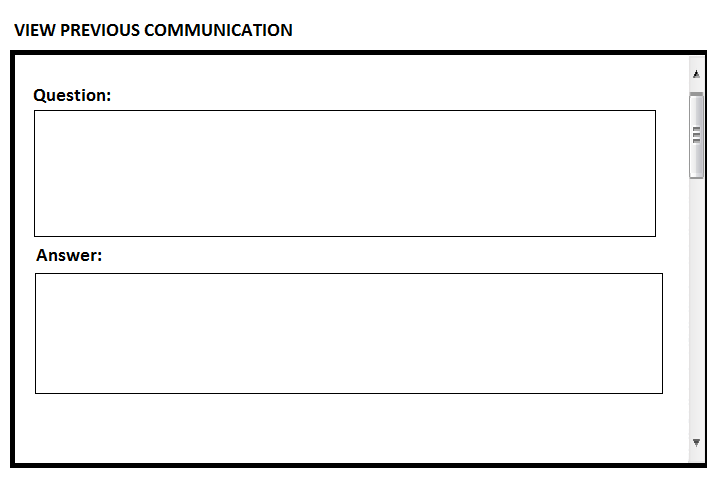
****

**Contact Module**

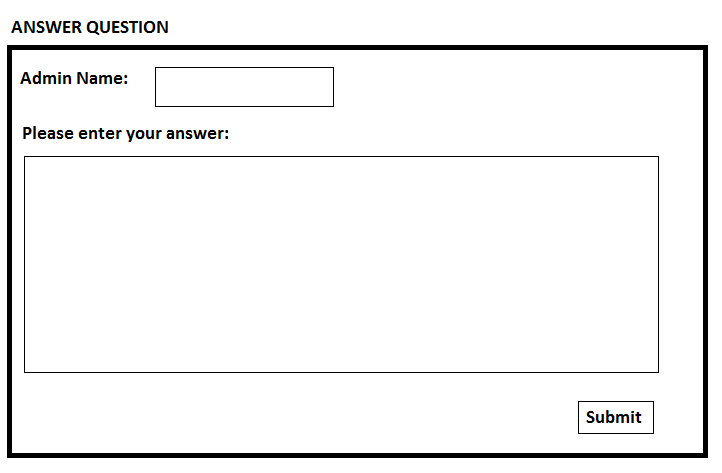
**Use case – Ask a question (user)**



**Use case – View previous communication (both)**



**Use case – Answer Questions (admin)**



**6.3 Screen Objects and Actions**

**Research / Buy Policy Module**

### Use Case: 1- Read More

Input: The user will select one of the coverages by clicking on read more under the specific type of coverage he/she is looking for.

Output: The coverage that been selected will open in a new page, with specific details about the policy and the protection it offers.

Action:

The user can click on read more button, which will take the user to a new page called insurance that has specific details about each coverage type.

### Use Case: 2- Configure

Input: User clicks configure under the coverage he/she interested in.

Output: A new page shows a form the user the deductibles, like cash recovery, and medical stuff.

## Action:

Upon the user click configure, the system will check if the user already has deductibles to view them.

### Use Case: 3- Buy Policy

## Input:

* Cash Recovery: A double representing the amount of money, and can’t be left empty or 0.
* Medical Stuff: A double representing the amount of money, and can’t be left empty or 0.

## Output:

Upon successful of submitting, the system will display a message declaring the purchase.

## Validation:

All entered data will be validated with the content from DB for current policies.

## Post Condition:

New and modified data will be saved into DB.

**Billing Module**

**Use case – Submit a Payment (user)**

**Input:**

First Name- String, minimum of 1 characters and maximum of 35 characters.

Last Name- String, minimum of 1 characters, and maximum of 35 characters.

Account Number - Int, minimum and Maximum of 10 digits.

Credit Card Number - String, minimum of 16 characters typed in standard credit card format.

CW2 Code – Int, minimum and maximum of 3 digits.

**Action:** When user submits, system will validates inputs for the set restrictions.

**Validation:** The system will check each field for correct input format. The system will also verify based on the account number if there is an existing account.

**Output**: If everything is entered correctly a confirmation message will sent. If there is any problems with validation restriction a message will appear to indicate the field there is a problem. If the system finds there is no matching account number it will give a message stating there is no existing account.

**Pre and Post Conditions:**

Precondition- User’s Account number is registered and exists in the system.

Post condition- If input succeeds then system will update database and create an invoice for the transaction.

**Description:**

The user wishes to submit a payment for a charge related to their insurance plan.

**Use case – Set Payment Plan (user)**

**Input:**

Account Number - Int, minimum and maximum of 10 digits.

Policy: String, User selects an existing Policy from drop down list. Default is none.

Payment Plan Selection: String, User selects Payment Plan from drop down list. Default is none.

Selection includes Monthly, 3 Months and Annually.

**Action:** When user submits, system will validates inputs for the set restrictions. Then sends user’s choice to database.

**Validation:** The system will check each field for correct input format. The system will also verifies if the user selected a Payment Plan.

**Output**: The page will display a confirmation message if successful. If unsuccessfully it will ask the user to reselect.

**Pre and Post Conditions:**

Precondition- User has an account. User has existing policies.

Post condition- If input succeeds then system will display messages and update the selection in the database.

**Description:**

The user wishes to set an automatic payment plan, rather than just paying charges as they are added to the account’s balance.

**Use case – View Invoices for all Policy (user)**

**Input:**

Account Number - Int, minimum and maximum of 10 digits.

Policies: String, User selects an existing Policy from drop down list. Default is none.

**Action:** When user submits, system will validates inputs for the set restrictions. Then sends user’s choice to database. It then returns all invoices for the selected policy in a table format.

**Validation:** The system will check each field for correct input format. The system will also verifies if the user selected an existing policy.

**Output**: The page will display a confirmation message if successful. If unsuccessfully it will ask the user to reselect.

**Pre and Post Conditions:**

Precondition- User has an account. User has existing policies.

Post condition- If input succeeds then system will display all the invoices but not change data.

**Description:**

The user wishes to see all current invoices on a policy. The user selects policy and the system returns all the invoices. The user can then further select an invoice and see individual information about that invoice.

**Use case – Update Invoice (admin)**

**Input:**

Account Number - Int, minimum and maximum of 10 digits.

Invoice Number- Int, minimum and maximum of 20 digits.

Updated Data - String, minimum of 1 words, and maximum of 500.

**Action:** The admin modifies the data displayed for the invoice and submits changes.

**Validation:** The system will check each field for correct input format. If correct it will send updated data to database.

**Output**: The page will display a confirmation message if changes are successful. If unsuccessfully it will ask the user to check their changes and resubmit.

**Pre and Post Conditions:**

Precondition- There are existing invoices to be updated.

Post condition- If input succeeds then system will display changes made to invoice and update data in database.

**Description:**

An admin wishes to make changes and update an invoice with new information. The Admin can modify the data displayed for the invoice. Such changes could be adding or removing charges.

**Contact Module**

**Use case – Ask a question (user)**

**Input:**

First Name- String, must be a minimum of 1 characters and maximum of 25 characters.

Last Name- String, minimum of 1 characters, and maximum of 35 characters.

Question- String, minimum of 10 characters and maximum of 100.

Code- String, minimum of 9 characters typed exactly as displayed.

**Action:** When user submits, system will validates input, if entered correctly a thread with question will be created, otherwise error message returned.

**Validation:** All required fields are entered correctly and security code is typed exactly as displayed.

**Output**: If everything is entered correctly no message will return, if fails system will return following message “invalid input”.

**Pre and Post Conditions:**

Precondition- User is registered and logged in.

Post condition- Successful submission will create a thread.

**Description:**

The user asks a question use case will allow the user to create a thread with a question for an admin to answer.

**Use case – View previous communication (both)**

**Input:** No input necessary.

**Output:** User will be redirected to previous communications.

Actions: System will redirect user or admin to all previous communication made between the two entities, being able to see all question and answers by simply scrolling down.

**Pre and Post Conditions:**

**Pre-condition:** User has logged in.

**Post condition:** System will redirect user or admin to all previous communication.

**Validation:** No validation is necessary since no input is required.

**Description:**

The view previous communication use case allow both admin and user to view previous communication between the two entities being able to see all questions and answers.

**Use case – Answer Questions (admin)**

**Input:**

Admin name- String, 1 characters minimum, and maximum of 20 characters.

Admin answer- String, minimum of 3 words, and maximum of 200.

**Output:** Admin will be redirected to answer and to the original question.

**Actions:** When admin hits submit a response to user question will be created.

**Validation:** System validates that input requirements are met, and creates a response to the question submitted by user.

**Pre and post conditions:**

**Precondition:** Logged in as administrator.

**Post condition:** No post conditions, response to question will be created.

**Description:**

The answer question use case allows admins to answer any question asked by the user.

**7. REQUIREMENTS MATRIX**

|  |  |  |
| --- | --- | --- |
| ***Module*** | ***Use Case*** | ***Design Component*** |
| ***Research/Buy Module*** | *?????* | *User Interface: ?????*  *ERD Diagram 1*  *Component Model: ?????*  *DB Schema section1* |
|  | *?????* | *User Interface: ?????*  *ERD Diagram 2*  *Component Model: ?????*  *DB Schema section2* |
|  | *?????* | *User Interface: ??????*  *ERD Diagram 3*  *Component Model: ?????*  *DB Schema section3* |

|  |  |  |
| --- | --- | --- |
| ***Module*** | ***Use Case*** | ***Design Component*** |
| ***Billing Module*** | *Submit a Payment* | *User Interface: Submit a Payment*  *ERD Diagram 1*  *Component Model: Billing*  *DB Schema section1* |
|  | *Set Payment Plan* | *User Interface: Set Payment Plan*  *ERD Diagram 2*  *Component Model: Billing*  *DB Schema section2* |
|  | *View Invoice for all Policies* | *User Interface: View Invoice for all Policies*  *ERD Diagram 3*  *Component Model: Billing*  *DB Schema section3* |
|  | *Update Invoice* | *User Interface: Update Invoice*  *ERD Diagram 4*  *Component Model: Billing*  *DB Schema section4* |

|  |  |  |
| --- | --- | --- |
| ***Module*** | ***Use Case*** | ***Design Component*** |
| ***Contact Module*** | *Ask question* | *User Interface: Ask question*  *ERD Diagram 1*  *Component Model: Contact*  *DB Schema section1* |
|  | *View previous communication* | *User Interface: previous comm*  *ERD Diagram 2*  *Component Model: Contact*  *DB Schema section2* |
|  | *Answer question* | *User Interface: Answer question*  *ERD Diagram 3*  *Component Model: Contact*  *DB Schema section3* |

**8. APPENDICES**

To be added… (Include any relevant information to the design of the project that you think will be necessary to the development of the modules.)