VISAGO

GRADE 12

IT PAT 2020

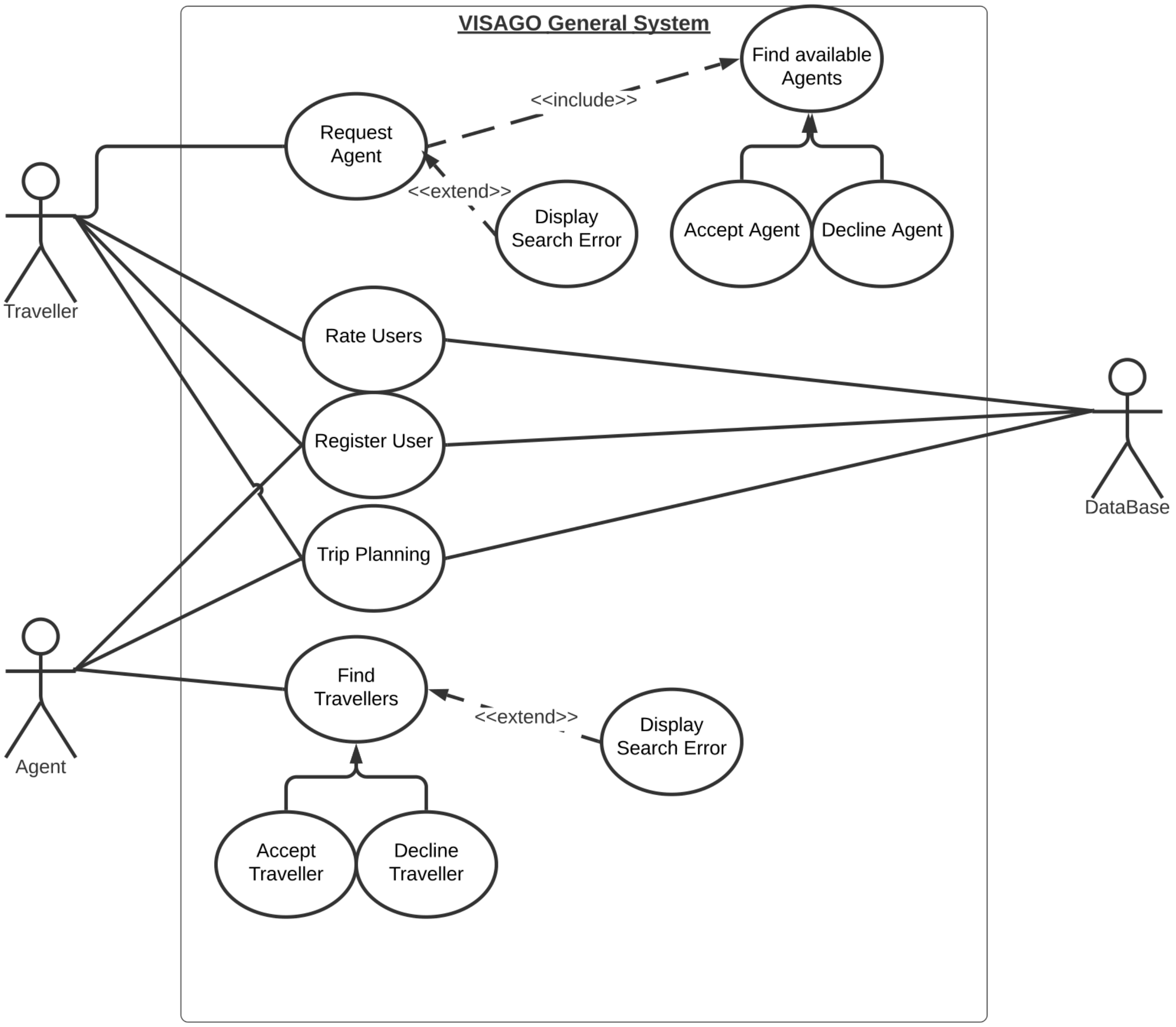
ANDREAS HUGH VISAGIE

TABLE OF CONTENTS

SCENARIO AND SCOPE

Imagine a situation where you and your friends/family want to go on an exotic trip to another country. So initially you would consult google for what would be the best deals for you to go to these desired destinations, but then you are unsure if you are getting a good deal or not. Now what if you could have an application that when you sign up, you are able to consult with people that are local to that country directly and plan your trip with these local people. People don't have much trust over the internet and are usually advised to be cautious when talking to strangers on the internet, that's why this application will need to verify and ensure prospective travellers that the person they are talking to about their travels is reliable and safe, hence the application will work on a peer-to-peer rating system. So let's outline what type of users this application should be catering for, so firstly we will need people that are local to these desired countries to assist those in desire of travelling, we will name this group of people "Agents". Secondly we will have people wanting to travel, hence we will name this group of people "Travellers". From this point on we will be referring to the two groups by their group names. The basic outline of this application is that when you sign up you will be faced with two options, Travellers or, Agents where you may select either one or both, then once registered the database will add the user based on the registration process. The Travellers will then be given the option to select a country and then specify what they would like to do in that country, then this data will be sent on to a stream where there are available Agents in that country. Agents may either accept or decline this Traveller help, once the Agent has accepted this Traveller, all data will be processed to the database and the Traveller will then be notified about this Agent, and a prompt message will show the Traveller the statistics of this Agent where the Traveller may ask either for a request or may accept this Agent. Once accepted both the Traveller and Agent will then be put in a chat room for planning and once they have an idea outlined they will enter the data into another form where the database will initiate a row in the Trips table. **Note***(*The trips table is where the planning for the trip will be stored. This will be further discussed in the design of the database.*)

USER REQUIREMENTS

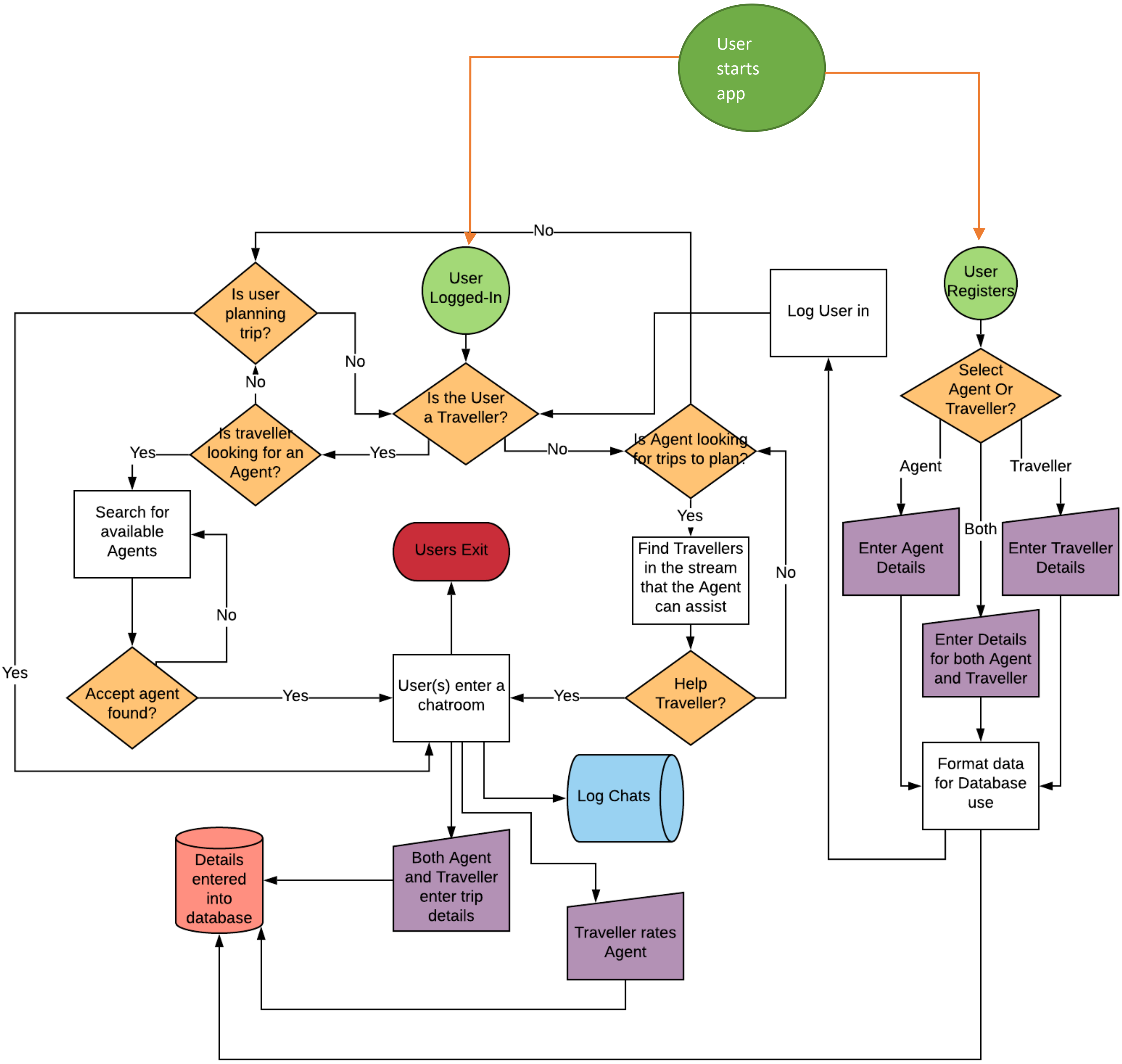


**(UML Use Case Diagram overview of the system/application)*

- *Agent may only register as a user and plan trips with the database.*
- *Agent Finds Travellers to help, the Agent may accept or decline Traveller.*
- *Traveller may register, rate Agent and plan a trip with the database.*
- *Traveller may find an Agent to help them plan a trip, the Traveller may accept or decline Agent*

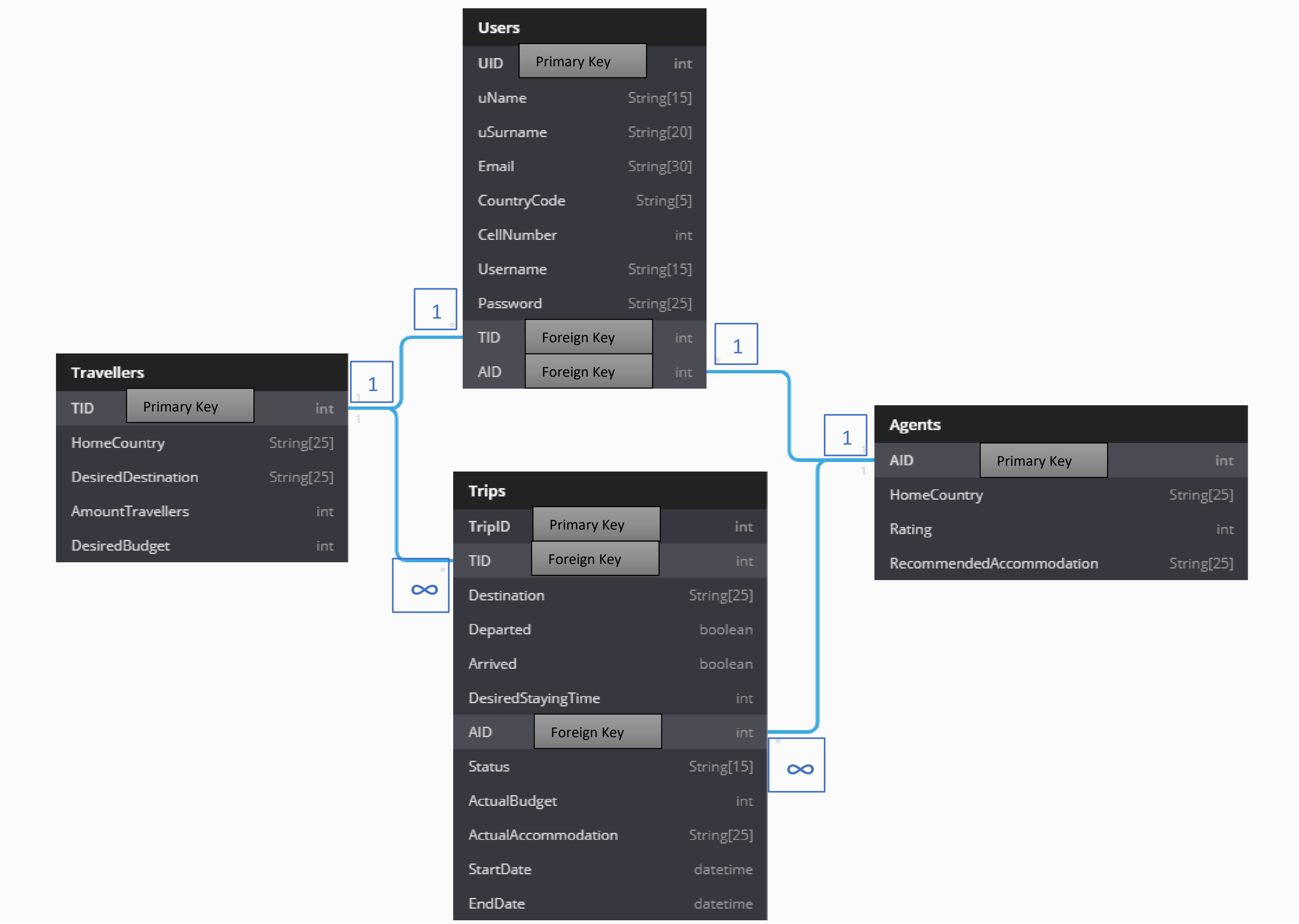
NAVIGATION

VISAGO global system overview


User starts app
User Logged-In
User Registers
Is user planning trip?
No
Is the User a Traveller?
Yes
Is traveller looking for an Agent?
Yes
No
Is Agent looking for trips to plan?
Log User in
Select Agent Or Traveller?
Agent
Traveller
Both
Enter Agent Details
Enter Traveller Details
Enter Details for both Agent and Traveller
Format data for Database use
Search for available Agents
Accept agent found?
Yes
No
Users Exit
User(s) enter a chatroom
Find Travellers in the stream that the Agent can assist
Help Traveller?
Log Chats
Details entered into database
Both Agent and Traveller enter trip details
Traveller rates Agent

- User starts by opening application, then will be faced with two options, “Login” or “Register”.
- Depending on what the user selects the flow diagram will follow to what the user selects at the initial screen.

DATABASE



Users		
UID	Primary Key	int
uName		String[15]
uSurname		String[20]
Email		String[30]
CountryCode		String[5]
CellNumber		int
Username		String[15]
Password		String[25]
TID	Foreign Key	int
AID	Foreign Key	int

Travellers		
TID	Primary Key	int
HomeCountry		String[25]
DesiredDestination		String[25]
AmountTravellers		int
DesiredBudget		int

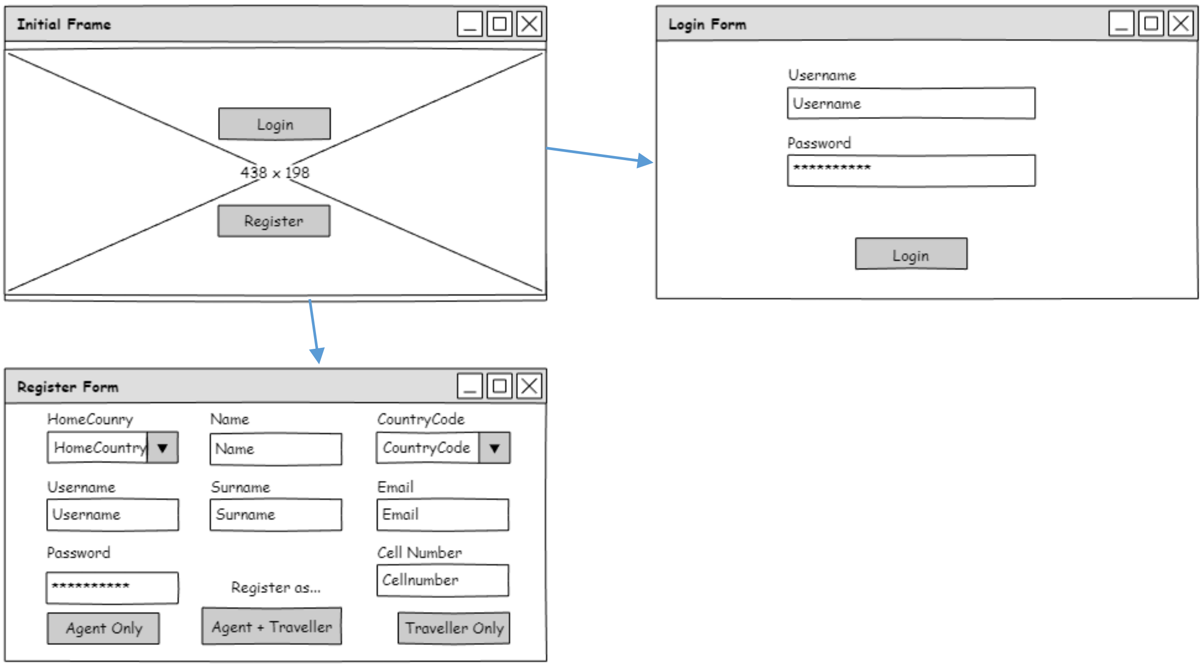
Agents		
AID	Primary Key	int
HomeCountry		String[25]
Rating		int
RecommendedAccommodation		String[25]

Trips		
TripID	Primary Key	int
TID	Foreign Key	int
Destination		String[25]
Departed		boolean
Arrived		boolean
DesiredStayingTime		int
AID	Foreign Key	int
Status		String[15]
ActualBudget		int
ActualAccommodation		String[25]
StartDate		datetime
EndDate		datetime

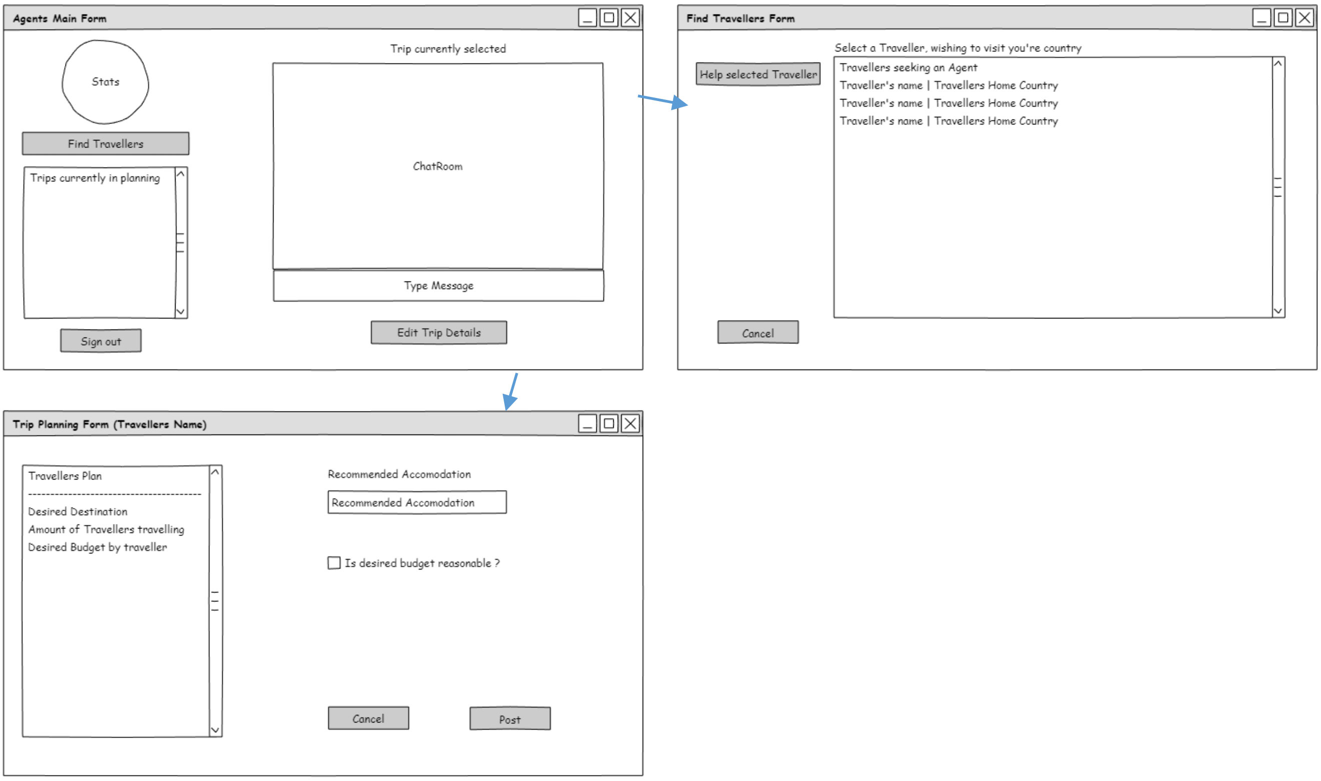
- *Travellers to Users has a one to one relationship*
- *Agents to Users has a one to one relationship*
- *Agents to Trips has a one to many relationships*
- *Travellers has a one to many relationships*
- *String [15] -> means that the datatype is of string and the field size is of 15.*
  E.g. "Status String [15] ".

GRAPHICAL USER INTERFACE (*GUI*)

Initial Forms (Figures A)

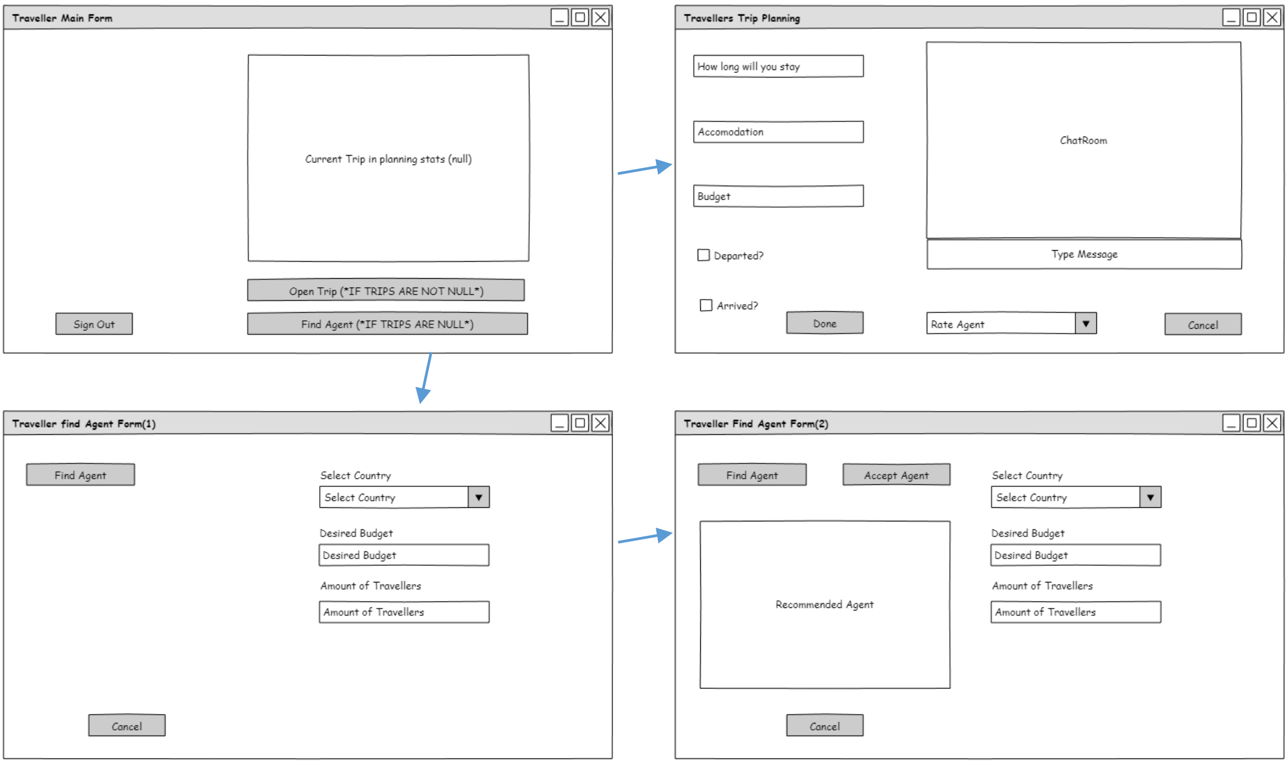


Agent's Forms (Figures B)

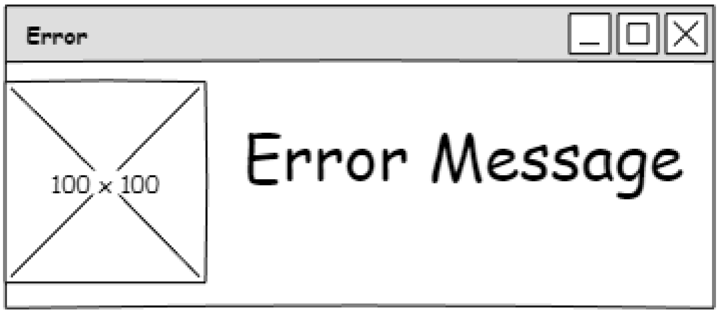

Agents Main Form
Stats
Find Travellers
Trips currently in planning
Sign out
Trip currently selected
ChatRoom
Type Message
Edit Trip Details
Find Travellers Form
Help selected Traveller
Select a Traveller, wishing to visit you're country
Travellers seeking an Agent
Traveller's name | Travellers Home Country
Traveller's name | Travellers Home Country
Traveller's name | Travellers Home Country
Cancel
Trip Planning Form (Travellers Name)
Travellers Plan

Desired Destination
Amount of Travellers travelling
Desired Budget by traveller
Recommended Accomodation
Recommended Accomodation
□ Is desired budget reasonable ?
Cancel
Post

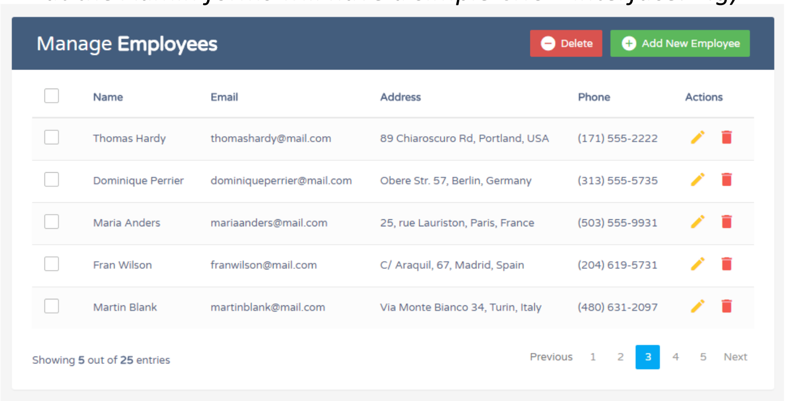
Traveller's Form (Figures C)



An Idea of what the Error messages will look like.



**(All Admin forms will be a part of the server which was not scoped for general user requirements, But the Admin forms will have a simple CRUD interface. E.g)*



Manage **Employees**

☐	Name	Email	Address	Phone	Actions
☐	Thomas Hardy	thomashardy@mail.com	89 Chiaroscuro Rd, Portland, USA	(171) 555-2222	✎ 🗑
☐	Dominique Perrier	dominiqueperrier@mail.com	Obere Str. 57, Berlin, Germany	(313) 555-5735	✎ 🗑
☐	Maria Anders	mariaanders@mail.com	25, rue Lauriston, Paris, France	(503) 555-9931	✎ 🗑
☐	Fran Wilson	franwilson@mail.com	C/ Araquil, 67, Madrid, Spain	(204) 619-5731	✎ 🗑
☐	Martin Blank	martinblank@mail.com	Via Monte Bianco 34, Turin, Italy	(480) 631-2097	✎ 🗑

Showing **5** out of **25** entries

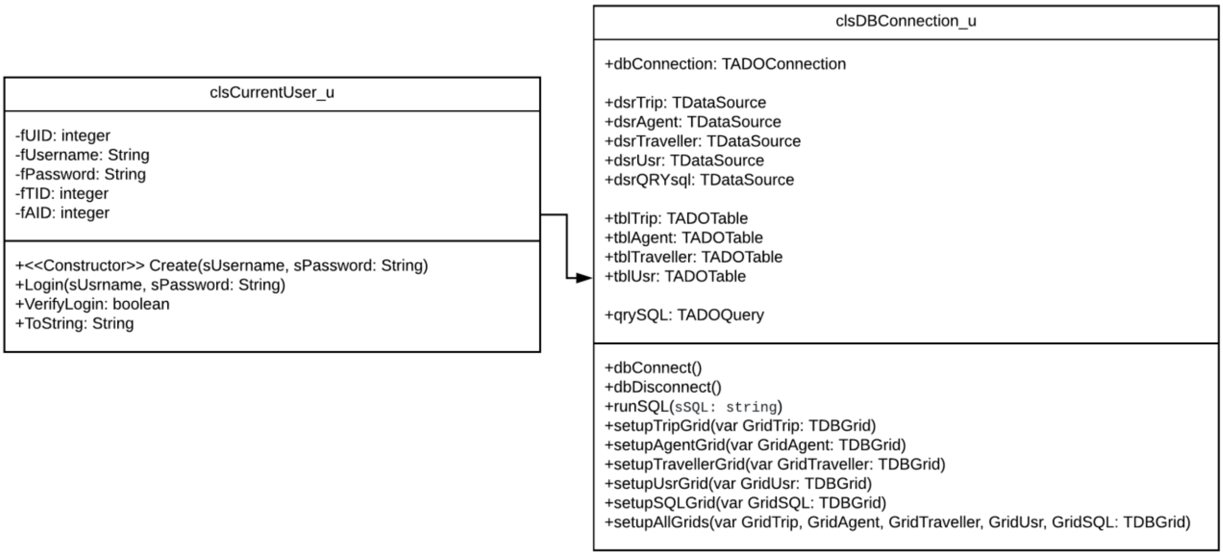
DATA DICTIONARY

Text Files – VISAGO will need a way of tracking logs in case the application or users run into any problems. This means VISAGO will be using text files to store logs, these logs will be updated frequently and VISAGO will only allow permission to super users to access these logs. Text files will also store countries with their codes.

Arrays – An Array will be populated by a text file containing countries with their codes. Another array will be used for storing Travellers, this will be used in the Agent side of the application.

Advanced – VISAGO will make use of sockets for internet protocols, so that online chatting may be possible, this means that a server will be necessary. The server will do some database interactions etc., hence it was not put in scope as this part is not a normal user requirement, this will be for developers and admins only as the server is very technical itself.

VISAGO's Classes



```
clsCurrentUser_u

-fUID: integer
-fUsername: String
-fPassword: String
-fTID: integer
-fAID: integer

+<<Constructor>> Create(sUsername, sPassword: String)
+Login(sUsrname, sPassword: String)
+VerifyLogin: boolean
+ToString: String
```

```
clsDBConnection_u

+dbConnection: TADOConnection

+dsrTrip: TDataSource
+dsrAgent: TDataSource
+dsrTraveller: TDataSource
+dsrUsr: TDataSource
+dsrQRYsql: TDataSource

+tblTrip: TADOTable
+tblAgent: TADOTable
+tblTraveller: TADOTable
+tblUsr: TADOTable

+qrySQL: TADOQuery

+dbConnect()
+dbDisconnect()
+runSQL(sSQL: string)
+setupTripGrid(var GridTrip: TDBGrid)
+setupAgentGrid(var GridAgent: TDBGrid)
+setupTravellerGrid(var GridTraveller: TDBGrid)
+setupUsrGrid(var GridUsr: TDBGrid)
+setupSQLGrid(var GridSQL: TDBGrid)
+setupAllGrids(var GridTrip, GridAgent, GridTraveller, GridUsr, GridSQL: TDBGrid)
```

SOFTWARE TOOLS

**(Figures A "Register Form")*

INPUT INTERFACES	INPUT VALIDATION	DATA PROCESSING	DATA OUTPUT
"Home Country" – The user will select a country from the combo box. The data type will be String and the combo box will be populated from an array. No specific format	If value not selected user will be prompted with a retry message. ***The retry message will prevent the user from progressing.***	The program will store this value in a String variable which will also be saved to the database.	Will be displayed in a memo for statistics. Store in database.
"Username" – User will type their username into the edit box, this data type will consist of String. No specific format	If edit box is empty prompt user with retry message.	The Program will store this in a String variable which will be processed to the database.	Store in the database. Will be displayed in a memo for statistics. Will be displayed to Travellers seeking Agents form.
" Password" – User types in password. Data type will consist of String. Format will be encrypted.	If edit box is empty prompt user with retry message.	Run encryption algorithm on String. Process encrypted data to the database.	Store in the database. No specific GUI output.
"Name" – User will type their name into the edit box, this data type will consist of String. No specific format	If edit box is empty prompt user with retry message.	The Program will store this in a String variable which will be processed to the database.	Store in the database. Will be displayed in a memo for statistics.
"Surname" – User will type their name into the edit box, this data type will consist of String. No specific format	If edit box is empty prompt user with retry message.	The Program will store this in a String variable which will be processed to the database.	Store in the database. Will be displayed in a memo for statistics.
"Country Code" – The user will select a country code from the combo box. The data type will be String and the combo box will be populated from an array. Format "+27" "+###"	If value not selected user will be prompted with a retry message.	The program will store this value in a String variable which will also be saved to the database.	Will be displayed in a memo for statistics. Store in database.
"Email" – User will type their Email into the edit box, this data type will consist of String. Format "abc@domain.com" "##@###.###"	If edit box is empty or the format is incorrect prompt the user with retry message.	The program will store this value in a String variable which will also be saved to the database.	Will be displayed in a memo for statistics. Store in database.
"Cell Number" – User will type their phone	If edit box is empty or the format is incorrect	The program will store this value in a Integer	Will be displayed in a memo for statistics.

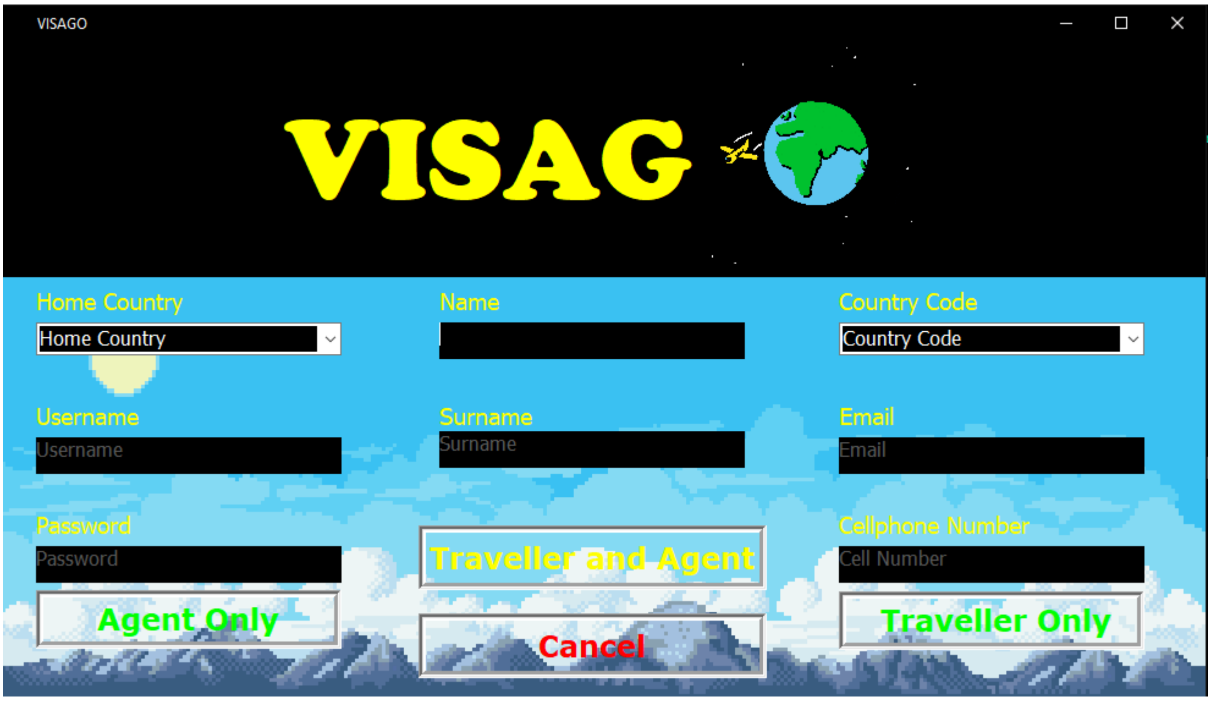
number into the edit box, this data type will consist of Integer. Format “013 555 1111” “##########”	prompt the user with retry message. Ensure Input is consisting of only Integer Values.	variable which will also save to the database.	Store in database.
“Agent Only” – User clicks button. No other input required	Prompt user with a verification message.	Generate an Agent ID only and process relevant information to the database.	User will be directed to Agent Form. Agent ID will be displayed in a memo.
“Traveller Only” – User clicks button. No other input required	Prompt user with a verification message	Generate a Traveller ID only and process relevant information to the database.	User will be directed to Traveller Form. Traveller ID will be displayed in a memo.
“Agent + Traveller” – User clicks button. No other input required	Prompt user with a verification message	Generate an Agent ID and a Traveller ID and process relevant information to the database.	User will be directed to a form where they will select which they mode would like to enter. All IDs will be displayed in a memo.

**(Figures C “Traveller find Agent Form (1)”)*

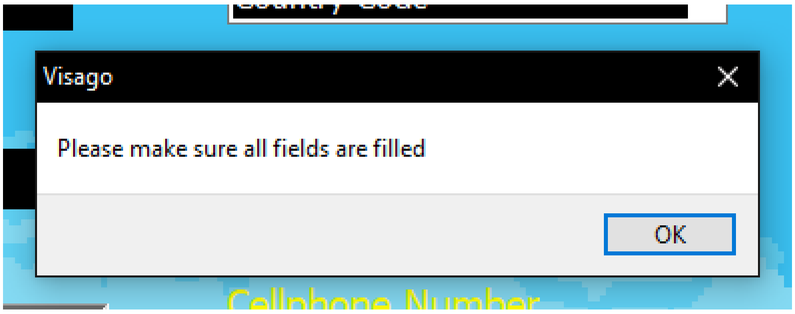
INPUT INTERFACES	INPUT VALIDATION	DATA PROCESSING	DATA OUTPUT
“Find Agent” – User clicks button. No other input required.	Prompt user with a search message.	Program will search the database for an Agent that meets the Selected Country field requirements. Create a memo.	The recommended Agent’s statistics will be outputted to the dynamically created memo.
“Desired Budget” – User will type their desired budget into the edit box, this data type will consist of Real/Float. No specific format	If edit box is empty prompt user with retry message. Ensure Input is consisting of only Real/Float Values.	This value will be added to the stream for the agent to view, it will also be processed to the database.	Will be outputted to the Agent’s Form in a memo for Travellers. Store in database.
“Amount of Travellers” – User will type their party size into the edit box, this data type will consist of Integer. No specific format	If edit box is empty is incorrect prompt the user with retry message. Ensure Input is consisting of only Integer Values.	This value will be added to the stream for the agent to view, it will also be processed to the database.	Will be outputted to the Agent’s Form in a memo for Travellers. Store in database.
“Select Country” – The user will select a country from the combo box. The data type will be String and the combo box will be populated from an array. No specific format	If value not selected user will be prompted with a retry message.	This value will be used to find an Agent in that country, it will also be processed to the database.	Will be outputted to the dynamically created memo.

REGISTER VALIDATION

- **Empty Field Validation**

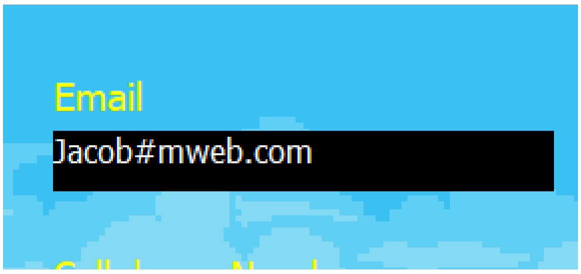


- If a field is left empty, the program will continue to prompt the user with an error message.



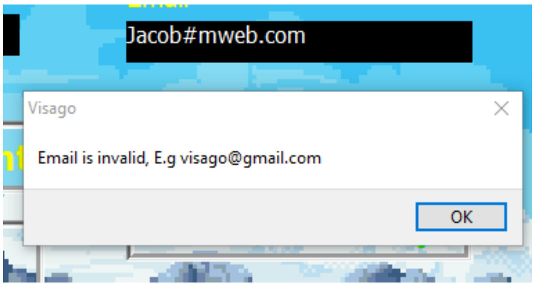
Please make sure all fields are filled

Email Validation



Jacob#mweb.com

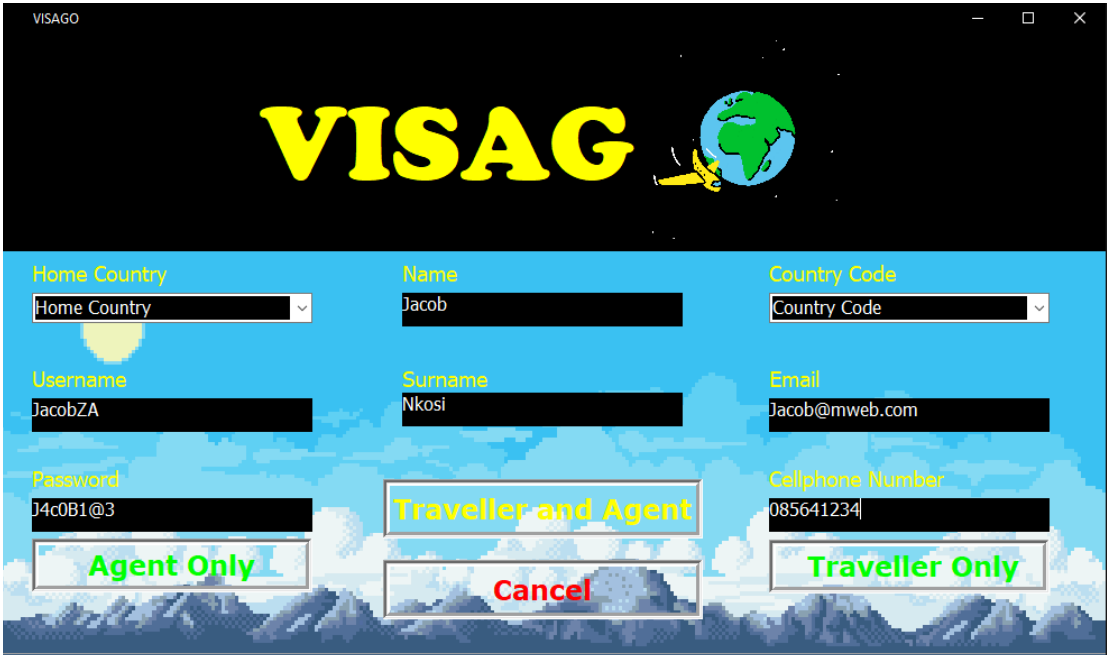
- Email must contain ‘@’ character.
- Email must contain ‘.’ Character.
- If email is incorrect, Prompt user with ‘Invalid Email’ error.



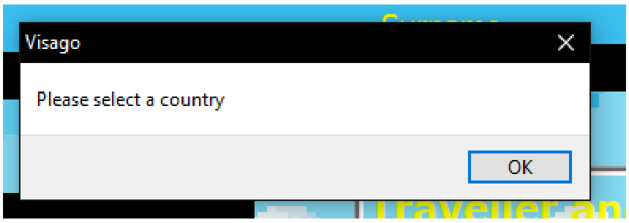
Jacob#mweb.com

Email is invalid, E.g visago@gmail.com

Selected Home Country or Country Code Validation



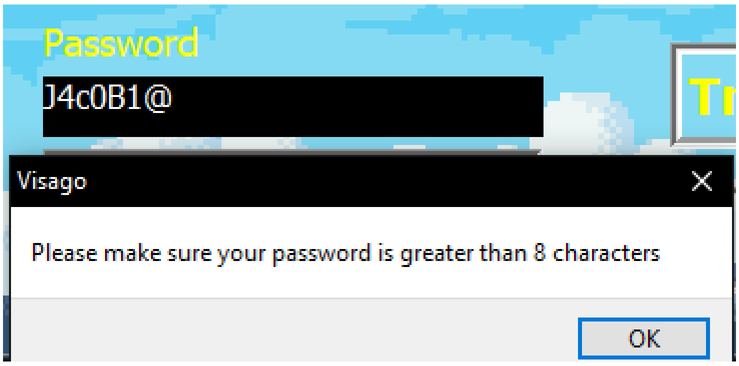
Home Country: Home Country
Name: Jacob
Country Code: Country Code
Username: JacobZA
Surname: Nkosi
Email: Jacob@mweb.com
Password: J4c0B1@3
Cellphone Number: 085641234



Please select a country

- If a country has not been selected, Prompt user with error.
- If a Code has not been selected, Prompt user with error.
- If a user selects a Country, the program will automatically select a country code correlating to that Country.

Password Validation

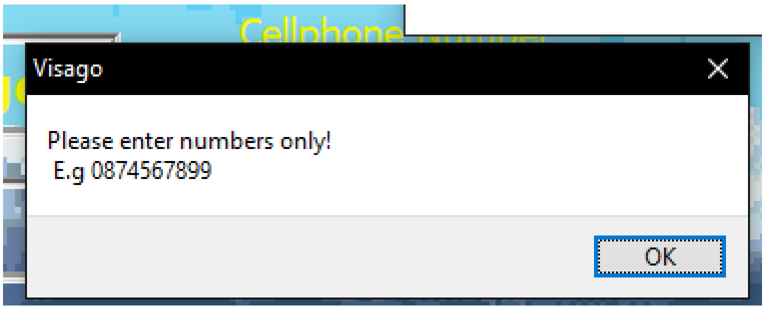


Password: J4c0B1@

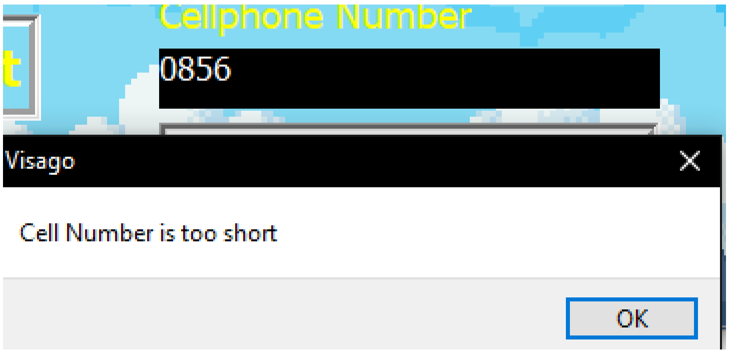
Please make sure your password is greater than 8 characters

- If password is less than 8 characters, prompt user with error message.

Cell Phone Validation



Please enter numbers only!
E.g 0874567899

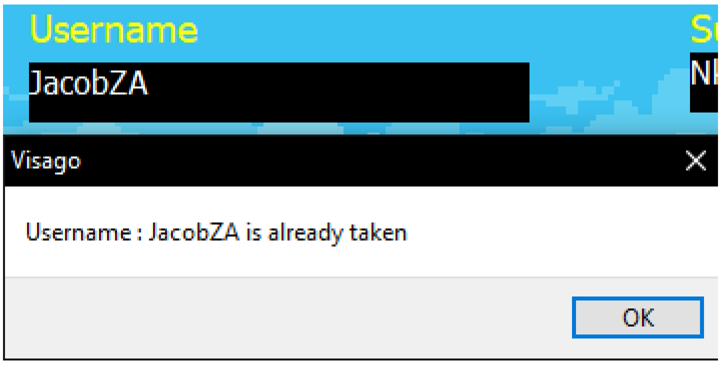


0856

Cell Number is too short

- If there is any other character other than numbers, the user will be prompted with an error message and the edit box will be reset to the value '0'.
- If cell number is too short, the user will be prompted with an error message stating that the number is too short (Number may not be less than 7 numbers).
- If cell number is too long, the user will be prompted with an error message stating that the number is too long (Number may not exceed 10 numbers).

User Existence Validation

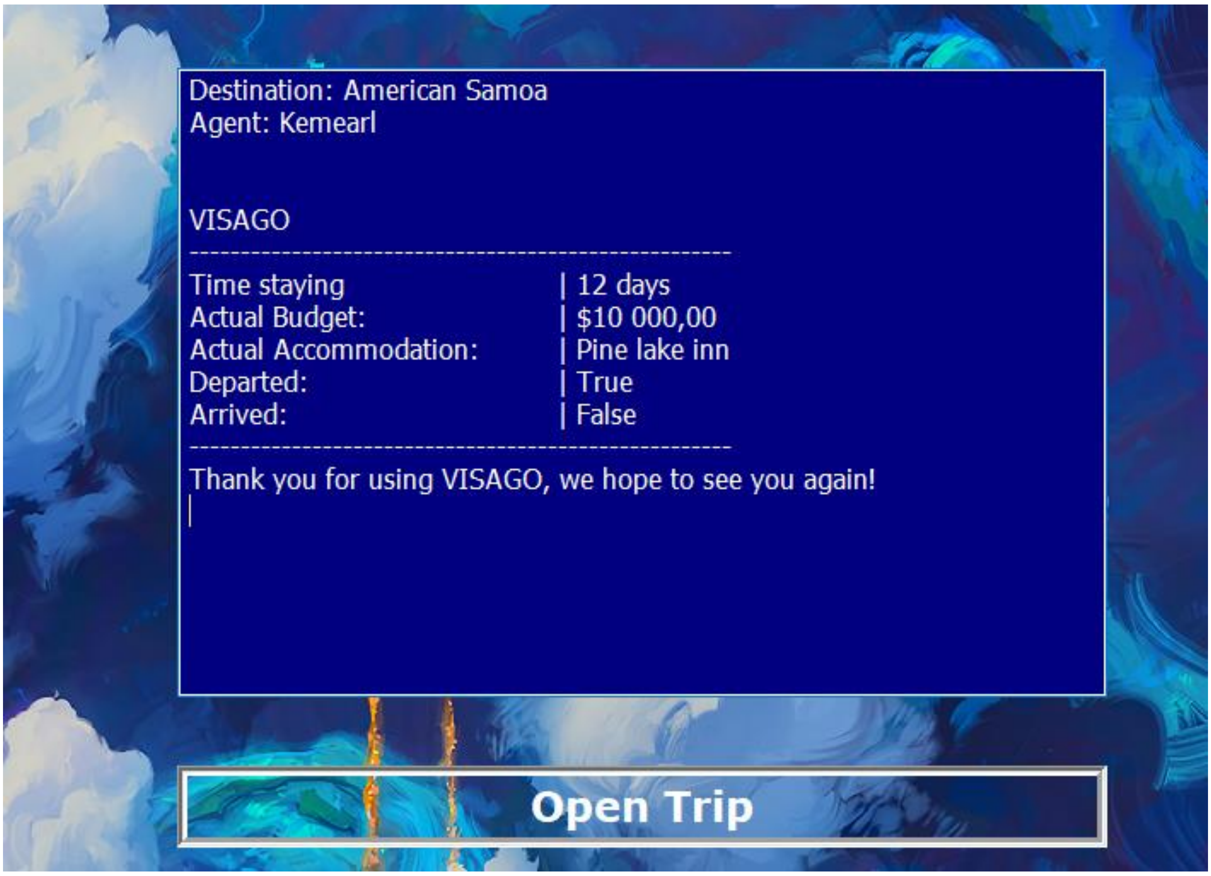


JacobZA

Username : JacobZA is already taken

- Check Database if this username exists.
- If this username exists, prompt user with error message.
- If this username does not exist, proceed to posting data to the Database.

OUTPUT FORMAT



```
Destination: American Samoa
Agent: Kemearl


VISAGO
----------------------------------------------------
Time staying                    | 12 days
Actual Budget:                  | $10 000,00
Actual Accommodation:           | Pine lake inn
Departed:                       | True
Arrived:                        | False
----------------------------------------------------
Thank you for using VISAGO, we hope to see you again!
```

- Once user has planned their trip, the program will format a slip for the user.
- The illustration above is an example of the formatted slip.