Group Contract

As a member of the Factorial Group, I agree to:

* Come to class regularly
* Attend all group meetings, barring emergency circumstances
* Notify the group of possible meeting conflicts as soon as they arise
* Check the GroupMe regularly, as this is the group’s primary means of communication
* Complete all assignments by their assigned due dates
* Ask fellow team members for help whenever necessary
* Help other team members whenever necessary
* Perform my fair share of the work

I certify that I have thoroughly read this contract and that I will abide by it. I am signing this contract at my own free will, and have initialed each of the above statements because I agree with it, and am willing to adhere to each clause.

I understand that breach of contract will result in a warning. If three warnings are accumulated I will meet with Armando and talk about my performance in the project.

This contract is subject to be amended pending a unanimous vote within the group.

Requirements Engineering

**User-** statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.

1. Allow multiple people to track their expenses
2. Keep track of separate trips
3. Log expenses, categorize those expenses
4. Finalize a trip
5. Print out a report

**System** - A structured document setting out detailed descriptions of the system’s functions, services, and operational constraints. Defines what should be implemented, so may be part of a contract between client and contractor.

1. Find platform that will be able to handle different users
2. Software must have a travel set with subsets of each separate trip the user went on
3. Software must compute sum of expenses, sort them out by category, and offer user a button that will finalize a travel.
4. Generate a pdf format report subject to the format of the Provost
5. Have a database maintain all data information for users, and allow for retrieval of that information in the future.
6. Provide an option that will summarize all the present information

**Functional-** Statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. May state what the system should not do.

1. System must handle separate users
2. System must keep track of separate trips
3. System must log expenses, categorize expenses, and option to finalize travel
4. System must generate pdf report of travel expenses on correct form
5. System must offer statistics option

**Non-Functional-** Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc. Often apply to the system as a whole rather than individual features or services

1. Platform must always accept new users and keep their information separate
2. Each user must store separate trips in a superset called travel
3. Expenses must be summed together for each trip, sorted into correct categories, and have a function to finalize a travel
4. Once travel is final, generate a report in pdf format
5. Implement database to store data of individual users, allow for retrieval of information
6. Documentation must be recorded throughout the duration of the project
7. Response time must be immediate
8. Reliability must be ensured

Technologies Roadmap

**Technologies to be used**

Front-end:

* Foundation and HTML

Database:

* MySQL

Server-side:

* Python based Django

**Execution Plan (For next two weeks):**

Front-End:

* Walk through a Foundation / HTML “Getting Started” Tutorial
* Familiarize ourselves with the documentation
* Build a starter app
* Research mobile responsiveness in a web app

Database:

* Relational Database Management System - Common choice for the storage of information in new databases. A program that lets you create, update, and administer a relational database
* Relational Database- collection of data items organized as a set of formally described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.
* Database Language: MySQL
* Data structure: the storage objects are tables which stored data in rows and columns
* 1. Produce a conceptual data model that reflects the structure of the information to be held in the database.

i. Develop an entity-relationship model, probably with drawings

ii. A Unified Modeling Language (UML)

* 2. Ask Provost questions detailing information he wants stored in program. Establish definitions of the terminology used for entities and their relationship and attributes.
* 3. Translate this into a schema that implements the relevant data structures within the database.

Server-Side:

* Going to walk through a “Getting Started” activity to learn more about Django and Python.
* Going to research server-side implementation to find what the most important aspects of it are and how to implement them
* Get a practice web framework up and running

Management:

* Organizing several meetings for the next two weeks, in an effort to improve group communication and set future goals
* Assisting other group members in any ways possible and checking in on each sub-team’s progress