SOEN-6751 HUMAN COMPUTER INTERFACE DESIGN

Instructor Dr. Rajagopalan Jayakumar

Assignment 1

Team members

Sr. No.	Name	Student ID	Email-id
1	Yash Chandreshkumar Golwala	40085663	golwalayash@gmail.com
2	Chirag Vora	40091454	chirag123192@gmail.com
3	Pooja Dhir	40104545	poojadhir1996@gmail.com
4	Harsh Divecha	40084737	harsh2826@gmail.com
5	Sanjana Udar	40094029	sanjana.udar@gmail.com
6	Jaswanth Banavathu	40080737	jaswanthbanavathu@gmail.com

Goal of project: The goal of the project is to design and implement a self-adjusting as well as self-learning smart interface for program development using GCC Compiler.

REQUIREMENTS GATHERING

Selecting the stakeholders

Stakeholder A person, group, or organization that is actively involved in a project, who can directly interact with the project and is affected by its outcome or can influence its outcome.

Facts about Stakeholder

- Name: The name of the stakeholders
- Company / Department: The department in which the stakeholder is involved
- Title / Role: The role or title which is played by the stakeholder
- Primary responsibility: The main or primary responsibilities of the stakeholder
- What tasks are you responsible for completing?
- To whom are you responsible for performing these tasks?
- What problems prevent you from performing your duties?

There are three different types of stakeholders:

I. Primary (Frequent User)

1. Tester (Someone testing the application)
2. Developer (Someone consuming the application)
3. Student (Someone learning programming)

The above three stakeholders are the **primary stakeholders** because these are the participants.

II. Secondary (Occasional)

1. Debugger: These are the people who try to find any bugs in the interface.
2. Professor: They use the interface to teach programming to the students in the class.

III. Tertiary (affected by)

1. Clients of Primary users: The people who use the interface in their organizations.

2. Competitors: People who develop the similar application. These people indirectly involve with the application.
3. Investors: They invest in the application by providing the required resources.
4. Open source community of developers: As GCC is an open source application; any people can access the interface.

Establishing the Requirements

Techniques:

Interviews: For exploration of the issues in the current system, interviews are the one of the easiest yet most powerful technique to get the requirements from the user. Though it is quite time consuming, it can be helpful for getting qualitative data.

Questionnaires: For getting answers for the specific questions, this technique can be used and can be helpful in generating the qualitative as well as the quantitative data. A specific set of questions are created, which is unambiguous to any set of users considering the same context to get the answers. This technique can be used to get more data from more and more people with low resource. Though it is powerful, it can be disadvantageous if users' responses are vague.

Focus Groups: This technique is used for collecting multiple perspectives from a same category of users. This will highlight the areas of conflict and will be easy to get the requirements from this data.

For covering the needs and wants of stakeholders and as anyone can conduct interviews if it is properly structured, we interchange the roles of being interviewer and interviewee with our group members to get the requirements from the developers' point of view. Also, to get the specific answers we used questionnaires and let a person who falls into categories of being novice, typical or expert user. We clubbed this information and did group discussion amongst our group and finalized the requirements list categorized as Functional and Non-Functional Requirements List.

Functional Requirements

Smart GCC interface has 2 windows

a. Program window: Showing the program user is working on.
b. Output window: Showing the output.

Post Installation: Ask user to select user type according to expertise.

Interface Initialization: Initialize the interface with menus and commands based on selected user type.

View All Commands: Interface for all users should have a menu option called “All Options” containing all the options that any user can select.

Execute Command: Once the user has selected the required command action from the menu, execute the program using selected action by calling GCC using the terminal available and display the result on the output window.

Interface is self-adjusting : The interface has a third window to show all the recent commands selected by the user and add the recently executed command if it is not listed in the “All Commands” list which was generated by default for that particular selected user type.

Non-Functional Requirements

Self-Adaptive: The interface should be adaptive to the type of user and to his preferences.

Responsive: The interface should be responsive, and it should not contain any delays of execution.

Fault tolerant: The application should not have any faults or errors in its execution.

Usability: It must be easy to use by different users on a daily basis.

Environment: The display and the environment of the interface should be satisfying the user.

Easy to use & Easy to learn: The interface should be easy to learn, and it should be easy to use for all types of users.

REQUIREMENTS DOCUMENTATION

User Characteristics

The most appropriate technique to document the characteristics of stakeholders is **Personas.**
The user characteristics illustrate attributes of the user, e.g. goal, motivations, frustrations, profile and his contact information and these will be documented by persona.

Developer

Persona for developer

CONTACT

adrianwals@gmail.com

+1-556-673-4847

2080, North Street New York City, NY.

Profile

Resourceful developer, very familiar with CSS, ASP, JS and Python. Enjoy problem solving and creating helpful, reliable user experiences for online tech companies.

ADRIAN WALS

Developer

Goal

Willing to learn new technologies in future. Wants to join a highly regarded software firm in a role leading proactive, dedicated teams.

Motivations

- Co-developed a dynamic, secure Web site from scratch. Launched visually appealing, user-friendly Webscape with interactive features to optimize traffic, page views, site "stickiness" and user experience (UX).
- Used SEO best practices to elevate organization's Web presence; achieved top 10 Google rankings on several highly competitive keywords.
- Created online surveys, contests and donation forms that boosted funding and organizational visibility. Helped drive $38K in ecommerce revenues, design and implement a self-adjusting smart graphical interface for C++ program development using the GCC compiler.

Frustations

- When running project stops in middle because of insufficient budget.
- When same basic thing has to be done again and again.
- When other developers don't contribute to project.

Student

Persona for student

CONTACT

aryanviyas@gmail.com

+1-456-973-9735

2080, Lincoln Avenue,
Montreal, Qc.

Profile

Current university student enrolled in software engineering with 1+ years of experience. Seeking to leverage solid skills in collaboration, communication and development as a programmer.

ARYAN VYAS

Student

Goal

Result oriented and innovative software engineer with 4 months of experience. Easily communicates complex technical requirements to others. Looking to join a highly regarded software firm in a role leading proactive, dedicated

Motivations

- Worked on Java platform using different frameworks available, learned about the industrial demands, skills set and work ethics.
- Worked as full stack developer, GUI designing and database administrator.
- Created custom plugins, templates, themes and modules in wordpress.

Frustations

- When software gives error and performance is very low.
- When advertisements comes during work.
- When response time is low.

Manager

Persona for manager

CONTACT

luciarose@gmail.com

+1-546-605-5345

103, Winifield, Charleston, SC.

Profile

Develop staff in line with strategic organizational and development objectives; coach others in value and methods of associate development; anticipate the changing demands for skills; select the best people for future; design processes for development of critical skill areas.

LUCIA ROSE
Manager

Goal

Wants to accomplish management-led objective in particular time period. To participate more in professional organizations or demonstrate certain skills or behaviors back on the job.

Motivations

- Achieved a 18% positive turnaround in growth by aggressively pursuing a 360 degree revamp of distribution, branding, pricing, and online strategies. Largest growth in Amazon (60%) and BJ's/Sam's (540%).
- Was one of only 4 categories in the company, out of 19, that achieved growth in a struggling international economy, and was the largest category in the company to achieve growth, with the largest total growth.
- Successfully spearheaded the launch of acquired brand NO/NO that generated $1.8 million in revenues with 8% growth at an average margin of approximately 55%.
- Won the 'Best Member's Choice Award' from Do It Best Corp. for design of a fly-through seed feeder.

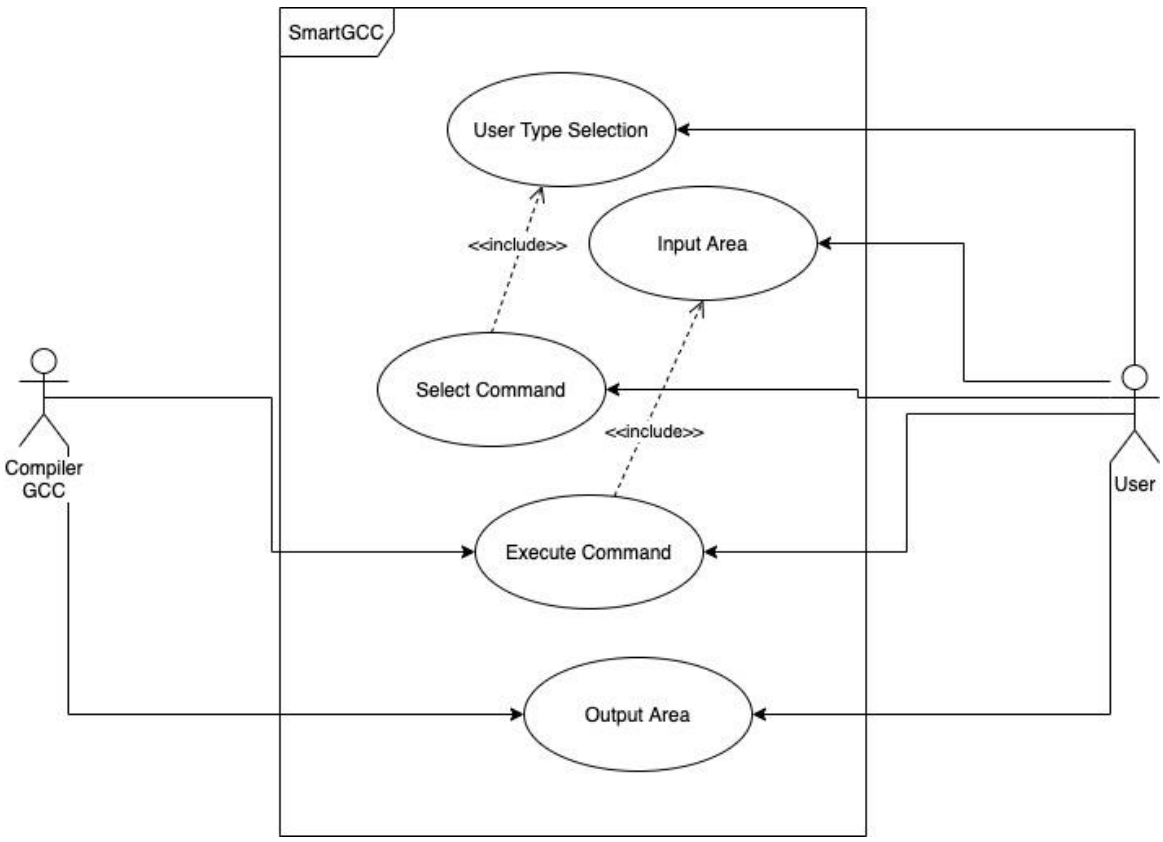
Frustations

- When internal politics drives into product scope.running project stops in middle because of insufficient budget.
- When senior leadership changes direction during product development.
- When developers demand time after deadline of submission.

Task characteristics using use case diagrams

The primary focus of use cases are on functional requirements of the project. The use diagrams consist of the tasks that are performed by the user. The use case represents a detailed view of the scenario and it is defined using a specification template.
The task characteristics are

1. When SmartGCC is open, the interface allows the user to select the type of user i.e., Novice, Typical and Expert.
2. The interface will initialize the interface with input and output tabs and will display the options as per the user type selection.
3. SmartGCC will execute the command selected or written by user through the available options and the result will be displayed on the command line.
4. The user can also select a different option from the “All Options” menu.
5. If the user selects a different option, the option is preserved in his history.
6. The additional option will be added to the user’s recent commands and it can be used regularly.



Use Case Diagram

Essential Use Case

The essential use case is the most appropriate technique to document the tasks because they represent the abstractions from scenarios and prevent the assumptions of traditional use case. It consists of three parts: a name that conveys the overall user intention, a step-by-step description of user actions and a stepped description of system responsibility.

USER INTENTION	**SYSTEM RESPONSIBILTY**
SmartGCC complier is installed	System prompting for user type selection
User selecting an option from All Options menu	System storing it in memory
User executing the program	Result must be displayed on the console

Task characteristics using hierarchical task model

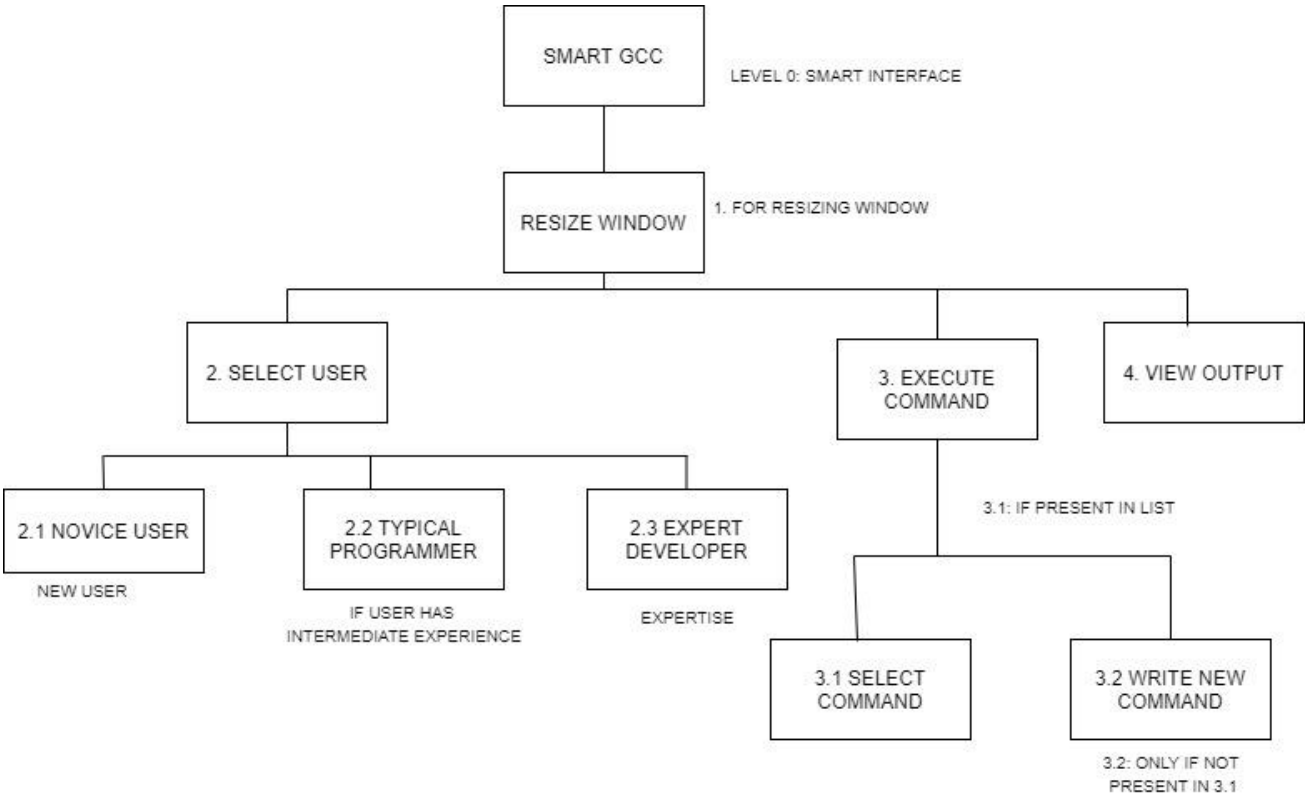
The hierarchical task diagram involves breaking a task into subtasks, then into another set of sub-tasks and so on. In HTA, we start with a user goal which is examined and the main tasks for achieving it are identified. The tasks are grouped as plans and they specify how the tasks might be performed in practice. HTA focuses on physical and observable actions, and includes looking at actions not related to software or an interaction device.

The diagrams are really helpful in recognizing the tasks and their subtasks in a graphical way which is easier to read and understand.

Plan 0 Smart GCC:

Plan View Output

1. If new user: 1, 2, 2.1/2.2/2.3, 3, 3.1/3.2, 4
2. if existing user: 3, 3.1/3.2, 4



Hierarchical Task Diagram

Narrative UI Scenarios

The narrative UI scenario is as follows

a. Nick Jonas being a student is trying to learn GCC. He opens the Smart GCC interface on his laptop. On the first page he gets an option to choose between which kind of user he is: 1) Novice, 2) Typical programmer or 3) Expert Developer. Here, as he is new to programming, Nick chooses the “Novice” option. It directs him to a screen which has an input area and an output area and various options like compiler options, linking options, execute options and debugging options. This way Nick is guided properly through the interface with the bare minimum that he’ll require to learn and get used to GCC.

b. Sam Mosby is a programmer by hobby and not completely unfamiliar with GCC. He chooses the “Typical Programmer” option under the type of user. He gets access to a few additional resources of code generation and code optimization options along with the options that Nick gets.

c. Nick can choose to select the “Typical Programmer” option when he opens the interface again sometime and is more comfortable with GCC. Similarly, Sam can choose to select

the “Expert Developer” option if he feels confident enough with GCC and wants to access the developer options in addition to the options in typical programmers.

Responsibilities

Module	Team_Member
Non-functional requirements & Narrative UI Scenario	Sanjana Udar
Establishing the requirements	Yash Chandreshkumar Golwala
Hierarchical Task Model	Chirag Vora
User Characteristics	Pooja Dhir
Functional Requirements	Harsh Divecha
Use Case Diagram	Jaswanth Banavathu

References

- Ics.uci.edu. (2020). https://www.ics.uci.edu/~kobsa/courses/INF231/07F/Ch_7_Presentation.pdf.
- "Flowchart Maker & Online Diagram Software", *Draw.io*, 2020. https://www.draw.io/