

Project 3

Requirements and Design Analysis

Prepared by:
Michael Lewellen
Spencer Stewart

Table of contents:

1. **Systems Concepts:** To make this report a stand-alone document, repeat the latest version of your system concept statement, as a synopsis of your project. Page 2
2. **Tailoring the Scope:** Describe how you made decisions to tailor the scope of this assignment for your own project and give justification where appropriate. Page 2
3. **Interaction Design Requirements:** Show about a dozen interaction design requirements that you extracted from your work activity notes and work activity affinity diagram. Page 2
4. **Requirement Extraction Process:** Briefly (one short paragraph) describe the requirements extraction process and how it worked for you. Page 3
5. **Models Used:** List the models you decided to construct for your project. Page 3
6. **Model Selection and Justification:** Justify why you picked these and why you did not make the others. Page 4
7. **Models:** Show each of your models. Label each with a sub-heading so we know what we are looking at. Page 4
8. **Additional Documentation:** Pictures of our initial ideas for models and application flow. Page 6

1. System Concept Statement

SnowTekBot will help Snow College students solve computer problems they are facing. If SnowTekBot can't solve the problem, it will direct them to the right place to have their problem get solved. He will also:

- Have a friendly interface to be more appealing to Snow College Students
- Be programmed to react to certain responses
- Help the user with specific computer problems and questions
- Guide the user to a specific Snow College department that specializes in solving their problem.
- Have a friendly response system that will help mitigate user frustration.
- Have step-by-step solutions for generic and common problems that students often come across.
- Quickly escalate questions beyond its scope.

2. Tailoring the scope

We started with the idea of tech assistance for Snow College students. As we continued to talk about the different ideas we had for your application, we would often think about the how to best serve the user. What actions and processes would be most useful for them. After discussing it, we decided to follow a more simplistic objective. That way our users will receive the best help using our application.

3. Interaction Design Requirements

Schedule becoming too complicated for the user to use. But also having specific features that certain users like.

Must provide an effective interface for users to interact with.

- We want the user to feel welcomed when they use the app. We also want the interface the feel seamless. If we develop an interface that is clunky and is difficult to use, people will be less likely to use our app.

Alpha version of program will be able to offer solutions for 3 basic problems.

- Through our contextual inquiry and analysis we realized that there are many different problems that people have with their electronic devices. We organized the problems and found out that there are a handful of problems that people face. In due time, we plan to expand our application to handle an increase of problems as well as questions that are more difficult. So for starts, we just want to work on 3 relatively simple problems and how to solve them.

App will contain a failsafe to elevate users to helpdesk assistance.

- We know that the SnowTekBot can't solve every problem that the users will submit to the SnowTekBot. So we want to put a failsafe into the app that will allow the user to elevate the question to get direct help from the Helpdesk, or when the TekBot can't solve the problem, it will automatically elevate the problem to the helpdesk.

Email a transcript of conversation with the SnowTekBot, to the user.

- To help the user solve their problem, we want to email a transcript of the conversation the user had with the bot, to the user. This way the user can look over the solution again and work with the solution.

Anonymous use is an option.

- A lot of the time people just want instantaneous answers. Quick and simple. To support this idea, we are trying to just give the user the information they want. We won't require them to make an account or even enter any personal information. We offer only an email address so that they can receive a transcript from the bot.

4. Requirement Extraction Process

We first looked at the fears and their needs/desires that our users were having. We wrote them all down and then we looked at potential solutions for each of these fears that they were having. We wrote down those ideas and determined which ones were most realistic to accomplish.

A lot of the solutions that were created were brainstormed in the middle of interviews and observations. These ideas were written down for later use.

5. Models Used

- Social Model
- Usage Model

6. Model Selection and Justification

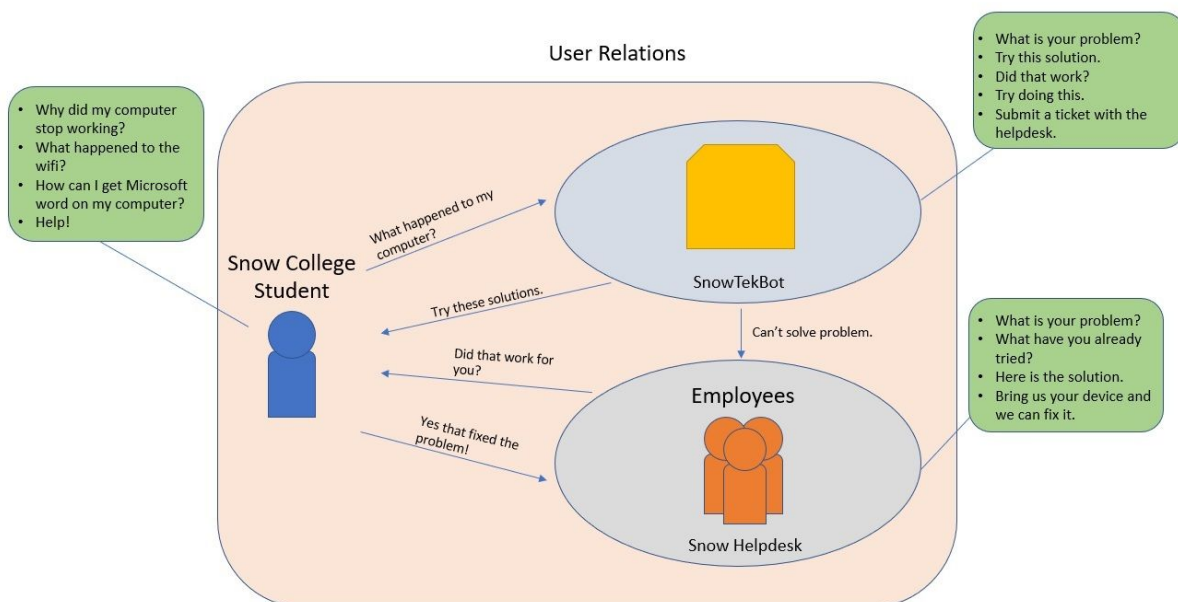
We decided to use the Social Model and the Usage Model. These two models we felt fit best with our target audience, Snow College students and faculty. They want to use SnowTekBot to be able to communicate their problems and get a solution, and most importantly they do not want to have to repeat themselves.

A major concern and desire amongst the individuals we interviewed, was to have the ability to get the right answer quickly. Our final model, of offering them access to multiple solutions will reduce the need to be repetitive and still be able to get the right answer quickly. The failsafe we build it to escalate their problem to the help desk. When looking at these functionalities, we decided that a usage model was appropriate to show the different features of the program. In this light, we realized that the people using it also had a lot of social connections as well. Using the social model to show the interaction between the primary user and the other groups.

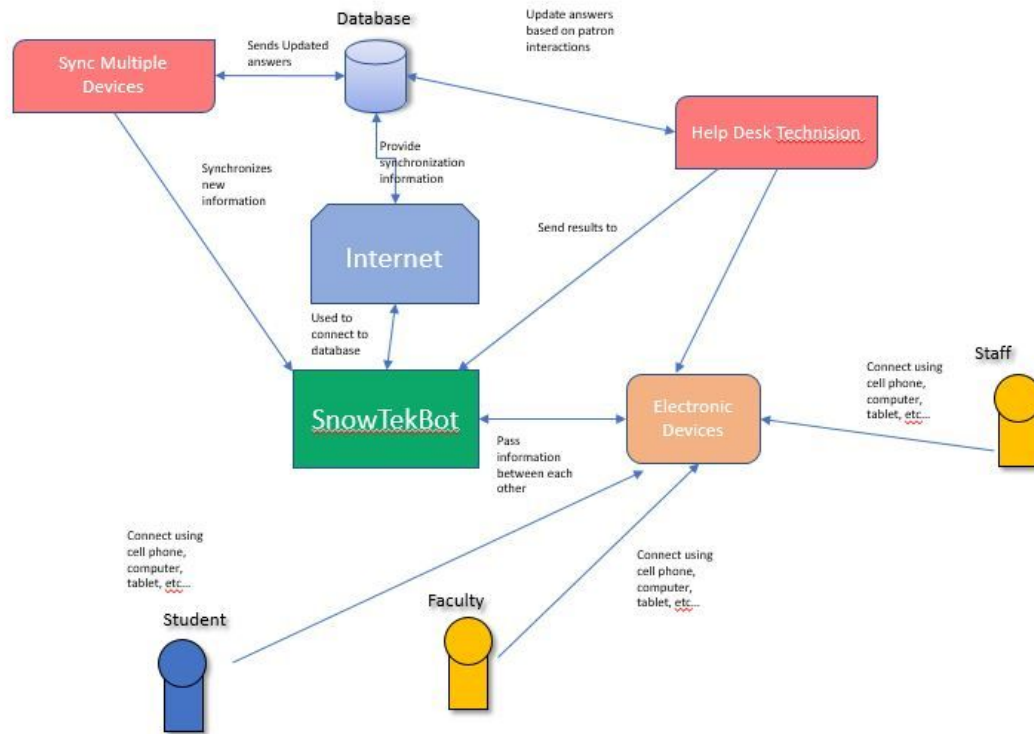
We felt that these two models would show the functionality and distinctness of the SnowTekBot more than any of the other models.

7. Models

Social Model



Usage Model (Flow Model)



The whiteboard features a hand-drawn flowchart for a chatbot named "SnowTex Bot". The flowchart is as follows:

- Welcome Screen:** A box containing a robot icon, the text "Welcome", a text input field labeled "email (optional)", another text input field labeled "Question", and a button labeled "Submit".
- Flow:** An arrow points from the "Submit" button to a central processing area.
- Processing Area:** A large box containing:
 - A sub-box with "Email" and "Phone" fields.
 - A box labeled "Ask Ticket URL" with a "Done" button.
 - A box labeled "Thank you" with a "Done" button.
- Confirmation Screen:** A box with a smiley face icon, the text "Thank you!", a "10 sec" timer, and the text "disappears and closes".
- Feedback Loop:** A box labeled "Confirm Helpdesk" with a "Yes" button and a "No" button. Arrows from these buttons point back to the "Welcome" screen.
- Annotations:**
 - "Exclude Done" is written in green above the "Ask Ticket URL" box.
 - "Confirming Complete" is written in green next to the "Confirmation Screen".
 - "Strong" is written in green next to the feedback loop arrow.
 - "Reply" is written in green next to the "Confirm Helpdesk" box.
 - "Edit Text" is written in green next to the "Confirm Helpdesk" box.

Requirements

- (1) MUST PROVIDE Effective Interface
- (2) Provide Solutions to 3 common problems
- (3) Fail safe SEND Help ticket w/ transcript
- (4) Email transcript to user if email provided
- (5) Anonymous Use is an option

Additional Information

<tags>

Example Reset Password forgot

10 Buzzwords per question.

I can't get my pw to work Help!

How do I reset my password?

FAQ

<How to Reset PW>

Confirm understanding

☒ Yes

☐ No

I can't get into Canvas!

- DPR1
- DPR2
- DPR3
- SUBMITTICKET