*What do I need to take notes on?*

|  |  |
| --- | --- |
| *What are some goals I found?* | *What are some objectives I found?* |
| Allow specific buildings/organizations to maintain control of their building | |
| Allow easy access for all | Allow appropriate people to see if a room is reserved or not |
| Allow viewing of ALL reservations | |
| Create a notification that works effectively with all of those involved in a reservation | |
| Create a system that doesn’t need of training but still has powerful features | |
| Show a vacancy list | |
| Create some kind of priority system for room reservations | |
| Create a scalable system | |
| Allow feedback on room conditions | |
| Allow users to see room assets and notify building managers to provide those assets | |
| Some institutions don’t want/don’t need outside user support | |
| Integration with current software/systems | |
| We want to keep cost to user and to use as to development low | |
| We want a consistent interface | |
| Influence the create a central system administrator | |
| Reserving a room should be quick | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Summary of evaluation questions and methods (how did you arrive at your design - contextual inquiry and analysis, requirements and design, prototyping and evaluation. Provide a description of your assess your success/failure during each stage of the process).

Key Findings:

* There are politics involved with room scheduling, including scheduling bias, unknown details when scheduling different events and different locations within an institution
* Current reserva
* tion system takes a lot of time and is very prone to bias/error
* A system of emails and phone calls is not working efficiently and sometimes reservations are missed or double booked.
* There is an existing system at Snow College that uses an Oracle and two front-end clients
  + It has an event manager
* Web based applications are popular and easy to access by all
* There are already lots of online scheduling systems
* Everyone likes to schedule a room at the same time/location
* A central system scheduling office/person is critical if this kind of software implementation was to work
* System can/should be integrated with currently implemented software
* Not everyone’s ideas work for the institution at hand - there needs to be a balance of ideas in order to try to satisfy everyone, especially the ones in charge and those that will be using the software product
* It makes people nervous to allow ANYONE to reserve a space - building managers don’t have to meet people over email/phone/in-person in order to reserve a room. It isn’t personable.
* A Work Activity Affinity Diagram (WAAD) was crucial to identifying key features and putting them into different sections/categories
  + It also allowed to find general themes for our software
* In-field sketches helped understand the existing system and how we can integrate our new software product alongside the current solutions/system
* The personas that are existing with the current system, proposed system and actual end result are drastically different from each other
* Modeling social and flow of software processes helps the developers and modelers how the system will work before it will be implemented. It allows a bridge to those going to the software before it is implemented.
* Our work roles don’t always capture the actual work roles
* Hand sketching works miracles - it is important to not do *everything* on the computer
* You can’t skip over ideation!
* Each user/work role does or must do something different than the designers expect
* User testing provides CRITICAL feedback that is used to improve design
* You cannot include everything in one version - cost is an important factor.

Important Result Information within the Project:

* We learned that many people want different features. There was big difference between want the administrator’s wanted and what the system users wanted.
* Software and ideals CHANGE with different circumstances, even during the current version of iteration
* Only specific models apply to a certain product solution - it is not necessary to use all types of models in order to design a project for the same user experience.
* It is important to review after each part of the design project and redirect focus and generate new ideals and models.
* The system concept statement must be versatile and easy to change.
* The thoughts of the designer/programmer/user must be fluid and understood
* Processes of software usage must be made *before* the design and software is created.
* You can’t skip over ideation!
* Design must be done in a certain way - steps must be followed in an order (mostly)
* Prototypes should be close to the actual implementation; testers will be expecting a similar interface
* Prototypes must have all of the ideas thought of in them, even if they don’t work as users will be looking for them.
* Steps of design may need to be improved or re-done in order to achieve the best design
* You cannot include everything in one version - cost is an important factor.

Key Successes:

* We found out some of the critical items that our software needed from our interviews and workshops at the beginning of our project - it was crucial part in creating our software product

Recommendations that address next project