Analytical Evaluation

For the analytical evaluation we are going to use Nielsen's heuristics to evaluate our UI. Using the guidelines Nielsen presents we will determine if we have a suitable UI. This evaluation should highlight areas that can be improved to allow for a better overall user experience. The specific heuristics we will be checking are listed below.

The tasks that we will evaluate are adding new courses to a student plan and modifying courses within an existing plan. We chose these tasks because they represent the main feature of the academic planning system. The whole purpose of the system is for EECS students to be able to create and modify their academic plan while enrolled in the College of EECS. These are the tasks that students will be spending most of their time on while using our system so we want to ensure these tasks are optimized as much as possible for a pleasant/efficient user experience.

Our team is suitable for doing this type of evaluation because we have received usability training via our CS352 course and now know what to look for as we do our evaluation. In addition, we are experts of the system interface since we are the ones creating it and have each had experience using the tool that is currently in place. We also have thorough understanding of the typical user of the system since we are EECS students ourselves and have been using it for years.

The user perspective we will take while doing this evaluation is the perspective of an OSU EECS student, either male or female. There is a wide age range of students, however the majority of students' ages range from approximately 18 to mid-twenties. The majority of these students also have extensive computer knowledge. This user perspective has been chosen since OSU EECS students are the target users of the UI we are creating.

Our program, for obvious reasons, requires a higher "usability" rating then many traditional programs. The need for this "high usability rating" stems from the user's frustration with the current scheduling interface. So, in attempting to attack this problem from two perspectives (analytical and empirical), we chose heuristic evaluation in order to evaluate by a set of rules that has no bias and evaluates critically in terms of usability. This test/assessment will occur simultaneously but completely separate from the empirical evaluation.

We will be using Nielson's heuristics for this evaluation. The questions that we will be asking ourselves are as follows:

- **Visibility of system status**: Is the system providing appropriate feedback at reasonable times?
- **Match between system and the real world**: Is the system providing "phrases and concepts familiar to the user, rather than system-oriented terms?" This includes real world conversations.
- **User control and freedom**: Does are system support a undo/redo system?
- **Consistency and standards**: Is the terminology that has been used consistent with the system and platform?
- **Error Prevention**: Does the design prevent a problem from occurring? If an error occurs, does the system provide a detailed enough message that related what went wrong with the program? If the user is able to use an action that may or may not cause an error, does the system ask for conformation before allowing the user to proceed?
- **Recognition rather the Recall**: Does the interface reduce the memory load of the user by making all of the objects, actions, and options visible when applicable?
- **Flexibility and efficiency of use**: Is it possible for the system to cater to both experienced and novice users through the speed at which the task can be completed? Can the user "tailor frequent actions"?
- **Aesthetic and minimalist design**: Is all the information that is displayed intended to be there; does the displayed information on the GUI have relevant qualities?

• **Help users recognize, diagnose, and recover from errors**: Are error messages “in plain language (no codes), [do they] precisely indicate the problem, and constructively suggest a solution”?

• **Help and documentation**: If help is needed with the interface, will the user be able to find the right help documentation and understand all concurrent instructions.

These questions will be used by two evaluators and be reported via a word document. At a later time period and upon the collaboration by the two evaluators, the findings will be combined into a single document with results that both evaluators can agree on.

The collected data that both collaborators will be gathering will be explicit and specific answers to Nielson's questions. These conclusions will then be a cornerstone in identifying problems with our interface. However, it is important to note that this data will not be the basis for major changes, but minor changes. Data that triangulates and agrees with the empirical evaluation will provide the basis for major changes to the interface.

**Empirical Evaluation**

We are trying to find problems with how our interface is designed, with specific interest in any problems that might be related to the learnability of the system.

The task we are going to evaluate is that of a student who is a Computer Science major on the Computer Systems track that has completed one term of classes at OSU and is planning the rest of their academic year. We've chose this scenario as it allows us to get an idea of how intuitive both the initial planning process and replanning processes are, based on the planning resources the system will provide.

We've chosen a male Mechanical Engineering student to perform the study on. We felt this student would be a suitable user in particular, because they have no knowledge of the existing processes the school of EECS uses for academic planning and so they won't be biased towards it. The user will have to be able to grasp what courses they need to take to advance their academic career exclusively from the resources we provide in the interface. We also felt this student was suitable because he has what we expect to be a standard level of ability in computer use.

For the Empirical Evaluation we are going to do a usability study using a paper prototype of our system. We've chosen this method of evaluation because one of our main goals is to make academic planning easier to learn, and the usability study should provide us with some insights on how well our interface achieves that goal. We chose the paper prototype for this as it is fairly easy to construct while still being a viable means of testing the interface. It also allows the user to get a somewhat interactive experience because we are using a dynamic paper prototype. It also allows us to avoid using a hi-fi prototype too early on in case there are any big changes.

Usability Studies produce a list of problems with the interface, so we will be collecting data based on problems users have with the interface. In our case, since this is a first time users will be performing the task, we anticipate getting a good set of data on learnability problems, along with data on other problems our interface might have.
Materials for Empirical Evaluation
Comm1110  Comm1140  ECE2710  WR1210

WR2140  WR2220  WR3270
<table>
<thead>
<tr>
<th>Fall 2009</th>
<th>Winter 2010</th>
<th>Spring 2010</th>
<th>Summer 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTH 112 (4)</td>
<td>MTH 251 (4)</td>
<td>MTH 252 (4)</td>
<td>MTH 231 (4)</td>
</tr>
<tr>
<td>COMM 111 (3)</td>
<td>WR 214 (3)</td>
<td>ECE 271 (3)</td>
<td>CS 160 (4)</td>
</tr>
<tr>
<td>WR 121 (3)</td>
<td>CS 161 (4)</td>
<td>CS 162 (4)</td>
<td>PH 211 (4)</td>
</tr>
<tr>
<td>PH 212 (4)</td>
<td>PH 221 (1)</td>
<td>PH 222 (1)</td>
<td>ECE 271 (4)</td>
</tr>
<tr>
<td>MTH 254 (4)</td>
<td>ST 314 (3)</td>
<td>COMM 111 (3)</td>
<td></td>
</tr>
<tr>
<td>MTH 112 (4)</td>
<td>MTH 251 (4)</td>
<td>MTH 252 (4)</td>
<td>MTH 231 (4)</td>
</tr>
<tr>
<td>COMM 111 (3)</td>
<td>WR 214 (3)</td>
<td>ECE 271 (3)</td>
<td>CS 160 (4)</td>
</tr>
<tr>
<td>WR 121 (3)</td>
<td>CS 161 (4)</td>
<td>CS 162 (4)</td>
<td>PH 211 (4)</td>
</tr>
<tr>
<td>PH 212 (4)</td>
<td>PH 221 (1)</td>
<td>PH 222 (1)</td>
<td>ECE 271 (4)</td>
</tr>
<tr>
<td>MTH 254 (4)</td>
<td>ST 314 (3)</td>
<td>COMM 111 (3)</td>
<td></td>
</tr>
<tr>
<td>Scheduled</td>
<td>Scheduled</td>
<td>Scheduled</td>
<td>Scheduled</td>
</tr>
<tr>
<td>Scheduled</td>
<td>Scheduled</td>
<td>Scheduled</td>
<td>Scheduled</td>
</tr>
<tr>
<td>Scheduled</td>
<td>Scheduled</td>
<td>Scheduled</td>
<td>Scheduled</td>
</tr>
<tr>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
</tr>
<tr>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
</tr>
<tr>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
<td>Passed</td>
</tr>
<tr>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
<td>Fail</td>
</tr>
<tr>
<td>Prereg. not completed</td>
<td>Prereg. not completed</td>
<td>Prereg. not completed</td>
<td>Prereg. not completed</td>
</tr>
</tbody>
</table>
COMM 111. PUBLIC SPEAKING (3).
Required - Taken
Prerequisites: None
Course Description: Public communication as it relates to informative and persuasive discourse. The theory and practice of public speaking in informative and persuasive contexts. Lec/rec. (Bacc Core Course)

COMM 114. ARGUMENT AND CRITICAL DISCOURSE (3).
Required – Equivalent Completed
Prerequisites: None
Course Description: Examination of argumentation as a part of human interaction and investigation. The course emphasizes the processes by which people give reasons to gain adherence and to justify beliefs and actions. The course includes readings, writing, and presentations concerned with the nature of arguments, processes of arguing, and argument criticism. Lec/rec. (Bacc Core Course)

CS 160. COMPUTER SCIENCE ORIENTATION (4).
Required - Taken
Prerequisites: None
Course Description: Introduction to the computer science field and profession. Team problem solving. Social and ethical issues surrounding use of computers. Lec/lab. PREREQS: Wireless laptop required.

CS 161. INTRODUCTION TO COMPUTER SCIENCE I (4).
Required - Not Taken
Prerequisites: MTH 112 or (MTH 251 or MTH 251H) or Placement Test
Course Description: Overview of fundamental concepts of computer science. Introduction to problem solving, software engineering and object-oriented algorithm development and programming. Lec/lab.

CS 162. INTRODUCTION TO COMPUTER SCIENCE II (4).
Required – Not Taken
Prerequisites: CS 161 and (MTH 231* or ECE 271*)
Course Description: Basic data structures. Computer programming techniques and application of software engineering principles. Introduction to analysis of programs. Lec/lab.
   • - Prereq may be taken prior to or simultaneously with this course.
CS 261. DATA STRUCTURES (4).
Required - Not Taken
Prerequisites: CS 162 and MTH 231

CS 275. INTRODUCTION TO DATABASES (4).
Required - Not Taken
Prerequisites: CS 261
Course Description: Design and implementation of relational databases, including data modeling, ER/UML diagrams, relational schema, SQL queries, normalization, user interfaces, and administration.

ECE 271. DIGITAL LOGIC DESIGN (3).
Required - Not Taken
Prerequisites: (MTH 251* or MTH 251H*) or MTH 231*
Course Description: A first course in digital logic design. Data types and representations, Boolean algebra, state machines, simplification of switching expressions, and introductory computer arithmetic. Lec/rec.
- Prereq may be taken prior to or simultaneously with this course.

MTH 111. COLLEGE ALGEBRA (4).
Required – Equivalent Completed
Prerequisites: MTH 095 or equivalent high school preparation.
Course Description: Polynomial equations and inequalities, polynomial functions and graphs, inverse functions, exponential and logarithmic functions, elementary mathematical modeling and applications. Lec/rec. (Bacc Core Course)

MTH 112. ELEMENTARY FUNCTIONS (4).
Required - Equivalent Completed
Prerequisites: MTH 111
Course Description: Triangle trigonometry, circular functions and graphs, trigonometric equations and identities, inverse trigonometric functions, polar coordinates, vectors and applications. Lec/rec. (Bacc Core Course)
MTH 231. ELEMENTS OF DISCRETE MATHEMATICS (4).
Required – Not Taken
Prerequisites: Placement in MTH 251
Course Description: Elementary logic, mathematical induction, sets, relations and functions, recurrence equations, algorithms.

MTH 232. ELEMENTS OF DISCRETE MATHEMATICS (4).
Required – Not Taken
Prerequisites: MTH 231
Course Description: Combinatorics, algorithms and complexity, graphs and trees. Lec/rec.

MTH 251. DIFFERENTIAL CALCULUS (4).
Required – Taken
Prerequisites: MTH 112
Course Description: Differential calculus for engineers and scientists. Rates of change: the derivative, velocity, and acceleration. The algebraic rules of differential calculus and derivatives of polynomial, rational, and trigonometric functions. Maximum-minimum problems, curve sketching, and other applications. Antiderivatives and simple motion problems. (Bacc Core Course)

MTH 252. INTEGRAL CALCULUS (4).
Required – Not Taken
Prerequisites: MTH 251
Course Description: Definite integrals, elementary applications to area, force, and work. Integral tables and basic techniques of integration, calculus of logarithmic and exponential functions, polar coordinates, applications to areas, volumes, force, work, and growth and decay problems. Lec/rec.

MTH 254. VECTOR CALCULUS I (4).
Required – Not Taken
Prerequisites: MTH 252
Course Description: Vectors, vector functions, and curves in two and three dimensions. Surfaces, partial derivatives, gradients, and directional derivatives. Multiple integrals in rectangular, polar, cylindrical, and spherical coordinates. Physical and geometric applications. Lec/rec.
MTH 306. MATRIX AND POWER SERIES METHODS (4).
Required – Not Taken
Prerequisites: MTH 252
Course Description: Introduction to matrix algebra, determinants, systematic solution to linear systems, and eigenvalue problems. Convergence and divergence of series with emphasis on power series, Taylor series expansions, convergence tests for power series, and error estimates for truncated series used in practical approximations. Lec/rec.

PH 211. GENERAL PHYSICS WITH CALCULUS (4).
Required – Not Taken
Prerequisites: MTH 251.
Co-requisites: MTH 252. Concurrent enrollment in a recitation section is strongly recommended.
Course Description: A comprehensive introductory survey course intended primarily for students in the sciences and engineering. Topics include mechanics, wave motion, thermal physics, electromagnetism, and optics. Elementary calculus is used. Laboratory work accompanies the lectures. Lec/lab/rec. (Bacc Core Course)

PH 212. GENERAL PHYSICS WITH CALCULUS (4).
Required – Not Taken
Prerequisites: MTH 252 and PH 211.
Co-requisites: MTH 254. Concurrent enrollment in a recitation section is strongly recommended.
Course Description: A comprehensive introductory survey course intended primarily for students in the sciences and engineering. Topics include mechanics, wave motion, thermal physics, electromagnetism, and optics. Elementary calculus is used. Laboratory work accompanies the lectures. Lec/lab/rec. (Bacc Core Course)

PH 213. GENERAL PHYSICS WITH CALCULUS (4).
Required – Not Taken
Prerequisites: MTH 254 and PH 212. Concurrent enrollment in a recitation section is strongly recommended.
Course Description: A comprehensive introductory survey course intended primarily for students in the sciences and engineering. Topics include mechanics, wave motion, thermal physics, electromagnetism, and optics. Elementary calculus is used. Laboratory work accompanies the lectures. Lec/lab/rec. (Bacc Core Course)
PH 221. RECITATION FOR PHYSICS 211 (1).
Required – Not Taken
Prerequisites: None
Co-requisites: PH 211
Course Description: One-hour weekly session for the development of problem-solving skills in calculus-based general physics. Lec/rec. Graded P/N.

PH 222. RECITATION FOR PHYSICS 212 (1).
Required – Not Taken
Prerequisites: None
Co-requisites: PH 212
Course Description: One-hour weekly session for the development of problem-solving skills in calculus-based general physics.

PH 223. RECITATION FOR PHYSICS 213 (1).
Required – Not Taken
Prerequisites: None
Co-requisites: PH 213
Course Description: One-hour weekly session for the development of problem-solving skills in calculus-based general physics. Lec/rec. Graded P/N.

ST 314. INTRODUCTION TO STATISTICS FOR ENGINEERS (3).
Required – Not Taken
Prerequisites: MTH 252
Course Description: Probability, common probability distributions, sampling distributions, estimation, hypothesis testing, control charts, regression analysis, experimental design.

WR 121. ENGLISH COMPOSITION (3).
Required – Not Taken
Prerequisites: Required of all students.
Course Description: Introduction to critical thinking, the writing process, and the forms of expository writing. Intensive writing practice, with an emphasis on revision. The term in which the student takes the course is determined alphabetically; see Schedule of Classes. (Bacc Core Course)
WR 214. WRITING IN BUSINESS (3).
Required – Not Taken
Prerequisites: WR 121 or Placement Test
Course Description: Continued practice in writing with an emphasis on the rhetorical and critical thinking demands of writers in business and industry. (Bacc Core Course)

WR 222. ENGLISH COMPOSITION (3).
Required – Not Taken
Prerequisites: WR 121 or Placement Test
Course Description: Continued practice in expository writing with an emphasis on argumentation and research. (Bacc Core Course)

WR 327. TECHNICAL WRITING (3).
Required – Not Taken
Prerequisites: WR 121 or Placement Test
Course Description: Continued practice in writing with an emphasis on the rhetorical and critical thinking demands of writers in scientific and technological fields. (Bacc Core Course) PREREQS: WR 121 or Placement Test