**Team Project Proposal**

**1. Provide a top-level, one-paragraph description of the problem to be solved by the system.**

Camp Ehawee, a summer camp in Mindoro, WI currently has a very convoluted and complex way of scheduling sessions and staff assignments. There are a lot of variables that go into the schedules: camper to staff ratios required by GSUSA, lifeguard ratios required by law, staff certifications/job position, session activity requirements, and staff break requirements. The camp needs an easy way to create and manage schedules for staff, sessions, and program areas.

**2. Give a one-paragraph description of the system you envision for solving the problem stated.**

We envision a system with an accessible user interface that, using registration info from campers and staff info from camp administration, designs schedules for both staff and campers so that all necessary requirements and regulations are met. The system will adjust for changing camper numbers and staff availability over time and will display current and relevant information about schedules to staff and customers of the camp.

**3. Name five examples of actors in the proposed system. Are actors (especially computer systems, databases, etc.) physically or logically distributed?**

|  |  |
| --- | --- |
| **Actor** | **Goals** |
| Camp Director | * Manually assign days off * Approve schedules * Design session curriculum * Manage/add rules and regulations for scheduling * Follow all rules & make sure staff also does so * (also all of the goals that the admin staff has) |
| Administrative Staff | * Manage camper registration information * Manage staff information * Manually adjust schedule * Manage staff preferences(for automatic scheduling and information purposes) * Stay up to date on session situations and requirements * Look at current and future schedules * See information about campers |
| Customer (camper/parent) | * See schedule for their session * Register for a session * Change registration information * Be up to date on any changes to schedule * Inform staff on camper needs |
| Specialty Staff (lifeguard, art director, etc.) | * See schedule for their program area * See information(size of the group, age range, theme, etc.) about sessions scheduled to be in their program area * See break schedule |
| Counseling Staff | * See schedule for their session * See information about campers in their session * See break schedule for their session * Switch breaks with another counseling staff |

Customers and employees are physically distributed because customers will be outside camp and all other people will be in the camp. Different employees are logically distributed by their duties in the camp.

**4. Describe, in 1-2 paragraphs, where the domain/business knowledge will come from (e.g., experts on campus, public online source, your personal expertise, "made up", combination of specific sources, etc.).**

Personal expertise and direct contact with Camp administration.

**5. Selecting a problem that is somewhat decomposable reduces risk. Describe, in 2-3 sentences each, at least two high-priority use cases, and at least two use cases that are desirable, but could be sacrificed depending on time constraints (i.e., medium priority).**

**High Priority:**

System generates schedule:

When camper registration and staff availability change, SuD adds new information into current schedule. If the schedule violate regulations, SuD designs a potential schedule that fulfills all given requirements and rules and notifies administrative staff about potential schedule. SuD waits for administrative staff to approval the potential schedule. If current schedule does not violate any rules, the schedule remains the same.

Camp Director manually edits/adjusts schedule:

Camp Director views current schedule. They use SuD to propose potential edits to the schedule and the SuD checks if the potential edits violate any rules or regulations. If the edits are reasonable, the Camp Director can save the changes to the schedule. If not, the Camp Director can continue to make adjustments to the potential edits, or revert to original schedule.

**Medium Priority:**

Counseling staff view information about campers:

Counselor wants to check if there’s any information they should know about the campers in their session. They use the SuD to view the session’s schedule, size, and any important information about campers(ex. allergies, diseases) that was entered during registration.

Counseling staff view schedule:

Counselor wants to see their schedule for their session. They use the SuD to view their schedule and are given the option to print the schedule.

**6. Supply a ballpark estimate of how long it would take 3 students (taking 2 other concurrent classes) to do two iterations of requirements analysis, design, and implementation to achieve an implemented, usable system with all high-priority and 50% of the medium-priority requirements completed. Justify your estimate in a few sentences. (Note that this estimate is not required to be 10 weeks or less!).**

17 weeks. The different constraints(ex. required employee ratios to campers depending on age) of the problem, scheduling across concurrent sessions, and automated schedule adjustments when registration information changes add a lot of complexity, so implementing the most basic version of the SuD will still be very difficult. Outside of the automated part of the SuD, the usability features and UI should be easier to implement.

**7. Details: Make a statement regarding what language(s), IDE(s), and version control software you have initially agreed to use as a software development team, and be prepared to justify your choices with regard to your proposed project.**

Language: Java, MongoDB

IDE(s): Netbeans

Version Control: git

Justification: Common previous expertise on Java, Netbeans and git. We choose MongoDB because of its flexible data storage API and dynamic structure which allows for rapid design implementation and exploration. Further, Jongo a Java plugin for MongoDB can map data records into Java objects for manipulation or JSON objects for sending to a web client.

Git’s decentralized design allows for both independent and collaborative development since projects can easily be branched and merged to match needs of the current state of development(ex. create many independent branches to explore viability of implementing different features during early development stages and focus on merging functioning features into cohesive project later). Additionally, the commit process allows for easy chronological documentation.