**Vision and Scope Document**

1. **Business Requirements**
   1. **Background**

The Computer Science department office makes Faculty Information Cards each semester, which are then put up outside each faculty member’s office, as well as the department office. These cards display course information, as well as office location, office hours, and contact information for the faculty member. This card is to inform students and others of availability for faculty members. Currently, the information is collected from the faculty members and then manually put into the system, by the office clerk, which then formats and prints them out individually.

* 1. **Business Opportunity**

The proposed system will provide an easier way to gather, collect, and compile the Faculty Information Cards to be printed. It will save time for the office member who is creating the cards by storing the information in a database, with an easy to use UI for modification, and export of Faculty Information Cards to be printed.

* 1. **Business Objectives**
     1. Reduce the time spent manually creating Faculty Information Cards.
     2. Decrease the difficulty and time spent updating faculty office hours.
  2. **Success Metrics**
     1. Decrease work time creating faculty information cards by at least 40%.
     2. Users with knowledge of the existing manual process can learn core functionality in less than 15 minutes.
  3. **Vision Statement**

For CSUS Office Administrators who need to print faculty information cards, Hornet CardGen is a database driven application that will provide a single point of access to faculty information. The product will store the relevant faculty information about faculty and staff, and allow for the data to be easily printed and posted for viewing outside faculty member and department offices. The database system will save Office Administrators time by providing an easy to use environment where they can easily view and modify faculty information. Unlike the current process of manually modifying the registrar’s file and manually formatting and creating the Faculty Information Cards, Hornet Cardgen will provide an easy user interface to modify the registrar’s file and once imported it will produce easy to print Faculty Information Cards.

In the current process, an Office Administrator (Reyna) gets an export from the registrar and manually formats and enters information into templates to be printed.

* 1. **Business Risks**

1. Since this product would be dealing with a database that possesses a vast amount of faculty information, it could be abused by malicious users.
2. This system would require information to be changed on a fairly regular basis, which would also require data consistency. In the event that information wasn’t changed or if the data is not accurate, it could result in cards printing incorrect information.
3. If the Registrar’s database system became non-operational or otherwise unable to export its data properly, Hornet CardGen would cease to provide value.
4. Since this product will be dependent on some third-party tools, there will inevitably be some maintenance cost. The system(s) serving the application may need monitoring and/or periodic updates. The Database Management System may also require some level of monitoring and maintenance. Without proper monitoring and maintenance, the system might be susceptible to failure.
5. If the Hornet CardGen project is not funded, then many valuable man-hours of the administration staff of CSUS departments will be expended on manually maintaining templates for Faculty Information Cards, a process which would greatly benefit from a software solution.
   1. **Business Assumptions and Dependencies**

Dependencies:

1. This system requires the external Registrar's system to be operational and accessible to Office Administrators for manual data export.
2. The format of the data exported from the Registrar system must be unchanged from the development phase of Hornet CardGen. If the format changes at all on the Registrar’s end, it is likely that the data import process in Hornet CardGen will be broken.
3. This system requires a database to store appropriate information and remove extraneous information imported from the registrar's import file.
4. This system requires an active printer to be accessible for the final task of printing the Faculty Information Cards.
5. This system requires an Office Administrator and Office Clerk to print the Faculty Information Cards and manage the information.
6. The system will export to a file that can be printed from most PCs. That file will be formated to print on Avery 5689 Cardstock.

Assumptions:

1. This first iteration of the system will primarily be used by the office administrator and office clerk in the Computer Science department. See Stakeholders section for more details.
2. In a future iteration, where more departments are considered to use the product, more analysis will have to be done to decide if any changes will be needed to accommodate them. For now, it is best to assume that there will be some amount of change needed. At bare minimum, time will need to be spent to train more people to use the product.
3. Appropriate resources will be provided for the system to be a success. These resources may include hardware and software that is required for everything from development to running and serving the application.
4. The external Registrar’s system will be assumed to be up and running, and the exported file given to Office Administrators does not change atfer Hornet CardGen undergoes development. This assumption should be followed up on and verified, should this project undergo actual design and development.
5. The system will require some level of maintenance for the Database Management System and possibly maintenance for a machine that will serve the application (pending application development and implementation).
6. **Scope and Limitations**
   1. **Major Features**
      1. Product will associate every faculty member with their related courses, sections, corresponding rooms and times, their office, contact information (phone or email), and their office hours.
      2. All of the information defined in feature 1 (excluding office hours) will be imported via a file manually downloaded from the Registrar’s external system.
      3. Office Clerks have an interface to add office hours for faculty members prior to printing Faculty Information Cards.
      4. Upon importing, the Office Administrator will have an overview of the imported information to look at. If the Office Administrator wants to make changes, the information should be able to be directly modified, or flagged to be changed in the future. This decision should be finalized in the design phase.
      5. Generates Faculty Information Cards in one operation. The output should be a PDF file, ready to be printed on Avery Form 5689 templated cardstock.
      6. Office Administrators can create, modify, and delete faculty any information.
      7. The clerk can create, modify, and delete office hours for their department’s faculty.
      8. The clerk can view all faculty information for their department.
      9. Office Administrators can reset another administrator’s password.
      10. Office Administrators can set up a department clerk for their department.
      11. Office Administrators can create new departments.
      12. Office Administrators can grant create a new office administrator, for their own or for newly created departments (departments with no Office Administrator).
      13. Office Administrators or Office Clerks can create Faculty Information Cards for their corresponding departments.
      14. There will be department privacy (Office Administrators and clerks will only see data related to their own department)
      15. Automatically maintains faculty office hours for a new semester, to be used in the event that there are no changes to a faculty member’s office hours.
   2. **Scope of Initial and Subsequent Releases**

|  |  |  |
| --- | --- | --- |
| Feature | Release 1 | Release 2 |
| Associates every faculty member’s entry in the database to related courses, sections, rooms and office number. | Each faculty member entry in the database will link to the faculty member’s related courses, sections, rooms and offices. | Fully implemented. |
| Import all data except office hours from a file generated by the external Registrar’s system. | The administrator can import faculty member’s office and class information into the database from a file downloaded from the Registrar’s system.. | Fully implemented. |
| Print Faculty information cards on Avery 5689 cardstock. | The administrator or clerk can print all the faculty information cards from a single command. Printing of individual cards is also permitted. | Fully implemented. |
| Create, modify, and delete database entries. | The office administrator can create new entries, modify entries, and delete entries within the database. | Fully implemented. |
| Enter or modify Office hours. | A department’s administrators or clerks can enter faculty member’s office hours to the database via the provided user interface. | Fully implemented. |
| View database entries. | A department’s administrators or clerks can view all the information in the database. | Fully implemented. |
| Reset passwords. | Office administrator can reset their department’s clerk’s password. | Office administrators can reset the password of other administrators. |
| Assign department clerk. | Office Administrators can assign their own clerk. | Fully implemented. |
| Grant Administrative privileges. | Not implemented. | Any office administrator can grant administrative privileges to another for their department’s database which doesn’t already have an office administrator. |
| User roles and permissions | Not implemented. | A department administrator or clerk can’t see another department’s database entries. |
| Upon creating a new semester, office hours from the previous semester (if available) will be auto populated for the new semester. This should allow for less data entry, should office hours remain the same for some faculty members. | Not implemented. | In creating a new semester, a copy of the previous semesters office hours will be copied and related to the new semester. Now if a faculty member lets the office clerk know that their office hours are the same, there is no change needed. |

* 1. **Limitations and Exclusions**

The only limitations to be aware of at this time is the lack of mult-department functionality in the initial release.

1. **Business Context**
   1. **Stakeholders Profiles**

* Office Administrator: Reyna Angeles, the Administrative Support Coordinator (for Spring 2017) for the Computer Science department at CSU Sacramento. The Office Administrator will presumably be the one that finalizes, OKs, and prints the contact cards for the department. They will need to interact with the system to some extent to do this. In future semesters, it is likely that Veronica Pruitt will be returning to the role of Administrative Support Coordinator for the department. As soon as other departments are considered for future iterations, other department Administrative Support Coordinators should be considered stakeholders as well.
* Office Clerks: The Computer Science department office clerks. The office clerks for Spring 2017 are Jessica and Andrew. These office clerks will be in charge of collecting the department faculty member's office hours and entering them into the system (or ensuring they are correct from a previous semester import). They will need to interact with the system to do that, as well as communicate with department faculty via email.
* Any and all project teams that will be working on future deliverables for the product. Of course these people want to succeed, and as the creators of the solution, should be empowered by the other stakeholders and key players to do so.
  1. **Project Priorities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dimension** | **Constraint** | **Driver** | **Degree of Freedom** |
| **Features** |  |  | Only single department functionality is required for the first release. |
| **Quality** | User interactions must be decreased by 40% relative to previous system. Users familiar with previous system should learn the new system within fifteen minutes. |  |  |
| **Schedule** |  |  | The Software Requirements Specification document is expected at the end of the Spring 2017 semester, and actual design will begin during Fall 2017. Therefore a first release could materialize as early as the end of Spring 2018 if another class built from the Software Design Specifications document from Fall 2017. |
| **Cost** |  | It is assumed that development tools and other required software are free, and development team are students. If non-free software is found to be needed, a proper budget should be negotiated. |  |
| **Staff** |  | Team consists of a product owner, a B.A. and a team of four to six CSC 131 students. These students will take on the role of writing a SDS. |  |

* 1. **Deployment Considerations**

This project will consist of both a server side and client side component. The server side component will need to house all the data in the system, and be accessible to the client side application. The client side application may also require some kind of hosting. This will depend on whether the design team decides on a web application or a desktop application. The software dependencies installed on the deployment server must be exactly the same as the versions used on the developer's test computers. In addition, the server(s) used should be set up accordingly to follow security best practices, so that the data cannot be accessed or manipulated by any third parties, and the system is not vulnerable to attacks.

Education in the basic use of the new system should take no longer than fifteen minutes for someone with experience working with the pre-existing system. This includes the reference of any training materials produced for this purpose, and as such the materials produced should reflect this.

1. **Estimation of Cost and Effort for delivery of the SRS**

Wages = $50/hr (assuming not free work through students)

Range of Hours = 155 minimum - 180 maximum

The minimum cost is $7,750 (assuming best scenario is 155 hours).

The median cost is $8,250 (assuming median is 165 hours).

The maximum cost is $10,800 (assuming worst scenario is 180 hours).

Our estimated cost for completing the SRS is 165 hours. If all work were not completed by students, you could assume an average rate of $50/hr for contractors, and see a cost of $8,250. This cost was estimated from the amount of time we expect to spend on each task in the product backlog. We have already completed (for the most part) business requirements notes and analysis, business rules notes and analysis, and now the Vision and Scope document. We used this as our basis for estimating the amount of time for the rest of the items in the product backlog. Because many of the key items in the Software Requirements Specification document or User Requirements document are related to the prior tasks, we expect that much of the considerations would already have been discussed by the time we start creating the User Requirements Document, or the Software Requirements Specification document (the presumed main key deliverables). Even though the key items in the Software Requirements Specification document will be discussed in prior tasks, we do expect there to be changes throughout the project, which means we will need to allow for time to incorporate those changes (including all prior documents). We are also accounting for changes to cost more time as we progress through the project, which is why the Software Requirements Specification document still has a fair number of hours dedicated to it. Our estimations make an attempt to take into consideration cases of misunderstanding requirements and unexpected emergencies, however, we weigh the chances of those events to be unlikely and therefore they do not have too much influence on the overall cost of delivering the Software Requirements Specification document .