BORS (Barangka Online Registration System)

Barangka National High School

Connect-Six

Vonn Niño D. Acosta

Kenneth Ian Gerard L. Alog

Mikael William E. Fuentes

Wesley Carl J. Ong

Andre Dominic D. Rondolo

Nigel Renzo C. Yu

May 14, 2018

Database Systems

CS 122A

Jessica O. Sugay

Table of Contents

1. Discussion of Implementation

1.1. Software Requirements

Our enlistment system will be web-based, so an Internet connection will be required to use it.
For our actual development, we used HTML for the structure of the webpage, and we used CSS for the visual layout of the webpage. We opted for HTML and CSS since they go hand-in-hand in webpage development. We also used PHP to connect the SQL server database to the webpage.
For the RDBMS, we used MySQL, since our classes covered it and we were more proficient in it.
Managing the database and web pages can be done on any OS.

1.2. Hardware Requirements

For users of the system itself, there are no hardware requirements other than those imposed by the operating system, since it will only require users to load a webpage. For deployment, a dedicated server is needed to run the system. The server should have the capacity to run for 24 hours a day. The server should also have ample disk space for the data. The server disk space would need to be at least 10TB large to hold all the student enlistment records. More disk space would be necessary in order to store multiple backup copies of the database in separate disks.

1.3. Limitations of the System

The system is a prototype and therefore is not developed enough for professional use. The prototype only has the following functions: login page functionality for students and admins, registration input by students, admin viewing of students, and admin adding of classes. Functions that are not yet implemented include: admin being able to delete student information, presenting a summary of student registration, editing of information, and viewing student's classes based on grade level.

1.4. Problems encountered during development

Many issues arose due to our inexperience with some of the programming languages required to implement the database and connect it to our webpages. We needed to choose and learn a language from scratch in order to connect our webpages to our database (we chose PHP over Django). We worked separately due to having to learn different languages, causing issues when attempting to implement them alongside each other (issues with variable names, miscommunication).

2. Actual Implementation

2.1. Tables and Integrity Constraints

Create database bores

```
DROP DATABASE bores;
CREATE DATABASE bores;
USE bores;
DROP TABLE STUDENT;
DROP TABLE EDUCATION_HISTORY;
DROP TABLE REGISTRATION;
DROP TABLE GUARDIAN;
DROP TABLE CLASS;
```

Create table Student

```
CREATE TABLE STUDENT(
        Learner_Ref_No INT NOT NULL PRIMARY KEY,
        First_Name VARCHAR(255),
        Middle_Name VARCHAR(255),
        Last_Name VARCHAR(255),
        Sex VARCHAR(6),
        Birth_Date DATE,
        Birth_Place VARCHAR(255),
        Land_Line VARCHAR(7),
        Email_Address VARCHAR(50),
        Cellphone_Number VARCHAR(11),
        House_Number VARCHAR(4),
        Building_Name VARCHAR(50),
        Street_Name VARCHAR(50),
        Village_Name  VARCHAR(50),
        Barangay_Name VARCHAR(50),
        City VARCHAR(50),
        ZIP_Code VARCHAR(5),
        Province VARCHAR(50),
        Country VARCHAR(50),
        CHECK ( Sex IN ('Male', 'Female'))
);
```

Create table Education History

```
CREATE TABLE EDUCATION_HISTORY (
        Learner_Ref_No INT NOT NULL PRIMARY KEY,
        Previous_School VARCHAR(50),
        Current_School VARCHAR(50),
        Grade_Level_Attainment INT,
        FOREIGN KEY (Learner_Ref_No) REFERENCES STUDENT(Learner_Ref_No)
        ON DELETE CASCADE
);
```

Create table Registration

```
CREATE TABLE REGISTRATION(
        Learner_Ref_No INT NOT NULL PRIMARY KEY,
        Date_of_Registration DATE,
        Grade_Level_Application INT,
        FOREIGN KEY (Learner_Ref_No) REFERENCES STUDENT(Learner_Ref_No)
        ON DELETE CASCADE
);
```

Create table Guardian

```
CREATE TABLE GUARDIAN(
        Learner_Ref_No INT,
        First_Name VARCHAR(255),
        Middle_Name VARCHAR(255),
        Last_Name VARCHAR(255),
        Sex VARCHAR(6),
        Relation_To_Student VARCHAR (50),
        Occupation VARCHAR(50),
        Email_Address VARCHAR(50),
        Cellphone_Number VARCHAR(11),
        FOREIGN KEY (Learner_Ref_No) REFERENCES STUDENT(Learner_Ref_No)
        ON DELETE CASCADE
);
```

Create table Course

```
CREATE TABLE COURSE(
        Course_Code VARCHAR(6) PRIMARY KEY,
        Course_Name VARCHAR(40),
        Grade_Level INT,
        Subject VARCHAR(40),
        CHECK( Subject IN ('English', 'Filipino', 'Math', 'Araling Panlipunan', 'Physical Education',
'Science'))
);
```

Create table Login

```
CREATE TABLE LOGIN_LOG(
        Username VARCHAR(255) PRIMARY KEY,
        Password VARCHAR(255)
);
```

2.2. Sample Queries

Checking data from a specific student

In this example, we check the data of the student with LRN 1

```
SELECT *
FROM STUDENT
WHERE Learner_Ref_No = 1;
```

Showing information of classes within a grade level

In this example, show information of classes within the grade level 7

```
SELECT *
FROM COURSE
WHERE Grade_Level = 7
ORDER BY Course_Code;
```

Showing a student's education history

```
SELECT EDUCATION_HISTORY.Learner_Ref_No, EDUCATION_HISTORY.Previous_School,
EDUCATION_HISTORY.Current_School, EDUCATION_HISTORY.Grade_Level_Attainment
FROM STUDENT, EDUCATION_HISTORY
WHERE  STUDENT.Learner_Ref_No = EDUCATION_HISTORY.Learner_Ref_No
ORDER BY STUDENT.Learner_Ref_No;
```

Showing the guardian information of a student

```
SELECT GUARDIAN.Last_Name, GUARDIAN.First_Name, GUARDIAN.Middle_Name, GUARDIAN.Sex,
GUARDIAN.Relation_To_Student, Guardian.Occupation, GUARDIAN.Email_Address,
GUARDIAN.Cellphone_Number
FROM GUARDIAN, STUDENT
WHERE  STUDENT.Learner_Ref_No = GUARDIAN.Learner_Ref_No
ORDER BY STUDENT.Learner_Ref_No;
```

Showing a summary after registration

```
SELECT STUDENT.Learner_Ref_No, STUDENT.Last_Name, STUDENT.First_Name,
STUDENT.Middle_Name, STUDENT.Sex, STUDENT.Birth_Date, STUDENT.Birth_Place,
REGISTRATION.Date_of_Registration, REGISTRATION.Grade_Level_Application
FROM STUDENT, REGISTRATION
WHERE STUDENT.Learner_Ref_No = REGISTRATION.Learner_Ref_No;
```

2.3. Final Logical ERD

2.4. Final Set of Screens and Reports

These are the final sets of screens that have been implemented in our system, thus all those that we're unable to implement are not shown here. The availability of the following screens depends on the role of the user. For example, a student can only register and view their own classes, while the administrator can view the student database, and view all the classes available to students.

Login Page

This is the very first screen the user is presented upon entering the website. To login, the user must input a User ID and a password. If the user is a new student applying to Barangka, the user can click "New student applying for Barangka?" link, which will direct them to the student registration form (which will be shown later).

Student Homepage

If the student logs in using a valid User ID and password, they will be redirected to the student homepage. The student can register for the next school year and view their currently signed up classes.

Student Registration

This screen can only be seen when the user clicks on either “Register” (from the Student Homepage) or the “New student applying for Barangka?” link (from the Login Page). The registration information screen is a form requesting the user’s student information (LRN, full name, gender, birthdate, birthplace, email address, cell phone number, and landline), address information (house number, building name, street name, village name, barangay name, city, zip code, province, and country), and guardian information (full name, gender, relation to student, occupation, email address, and cellphone number).

Confirm Registration

Clicking “Submit” from the Registration Form will bring the user to this confirmation screen. Clicking “Confirm” will return the user to the previous page they were on (either Login or Student Homepage).

View Student Classes

This screen is accessed by clicking “View My Classes”. After a student logs in with their credentials, they may use this page to see their currently enrolled classes.

Admin Homepage

If the user logs in using an administrator username and password, they will be redirected to the admin homepage. The administrator can view the database containing student information and the classes offered.

Student Database

This screen is accessed after the admin enters a valid set of login credentials and selects “View Student Database.” Here, the admin can view the list of students and whatever information they have input into the database.

Student Classes

This screen is accessed after the admin enters a valid set of login credentials and selects “View Classes”. Here the admin can view the current existing classes students can take.

Creating a New Course

This screen is accessed when they select “Create New Classes” via the Student Classes page. Here, the administrator fills out a course information form which asks for a course code, course name, grade level, and subject.

Confirming Class Creation

Clicking “Submit” from the Course Registration Form will bring the user to this confirmation screen. Clicking “Confirm” will return the user to the Student Classes page, where the new class has been added.

3. Appendix: Documentation of Group Meetings

MEETING DOCUMENTATION: Project Deliverable 3

MEETING DETAILS			
Date	04-19-18	Start Time	10:00
Location	F-228	End Time	10:40
Documented by	Alog, Kenneth		

AGENDA
- Discuss Project Deliverable 3 - Conceptualize the prototype of the program - Find system requirements for the prototype - Plan how to convert EERDs to mySQL language - Assign action items

ATTENDANCE (P=Present / A=Absent / L=Late Arrival / E=Early Departure)			
P	Acosta, Vonn	P	Ong, Wesley
P	Alog, Kenneth	A	Rondolo, Andre
P	Fuentes, Mik	A	Yu, Nigel

DISCUSSIONS	
Topic / Discussion Point	**Action Items**
System and Hardware - The group discussed the specs of the database system that can be implemented by the school	- More research on possible hardware that can be afforded by the school
Review Previous Deliverables - The group looked back at the specifications that we enumerated on the previous deliverables	- Learn how to implement the group's proposals - Research on what coding languages we will need for the project

ACTION ITEMS		
Action Item	**Point Person**	**Due Date**
- Research on what coding languages we will need for the project	Ken, Vonn, Andre and Mik	April 26
- Learn how to implement the group's proposals	Vonn and Andre	April 26
- More research on affordable hardware	Wesley and Nigel	April 26

NEXT MEETING					
Date	April 26, 2018	Time	9:30am	Location	F-228
Agenda	Project Discussion				

MEETING DOCUMENTATION: Project Deliverable 3

MEETING DETAILS			
Date	04-26-18	Start Time	10:05
Location	F-228	End Time	10:30
Documented by	Alog, Kenneth		

AGENDA
- Discuss Project Deliverable 3 - Conceptualize the prototype of the program - Find system requirements for the prototype - Plan how to convert EERDs to mySQL language - Assign action items

ATTENDANCE (P=Present / A=Absent / L=Late Arrival / E=Early Departure)			
A	Acosta, Vonn	P	Ong, Wesley
P	Alog, Kenneth	P	Rondolo, Andre
P	Fuentes, Mik	A	Yu, Nigel

DISCUSSIONS	
Topic / Discussion Point	**Action Items**
- Wesley and Vonn Discussion of Implementation – April 28 - Wesley Final Logical ERD – before May 14 - Andre and Vonn MySQL Commands – May 1 – Output: Code - Nigel and Ken Front-End – Draft by April 28 (Web Page Format), Learn PHP by May 1 - Ken and Mik Connecting Front End and Back-End – Learn by April 30 - Google Drive for documenting project Mik – April 26 - Fake Data Mik and Ken – April 30	- Please update
Review Previous Deliverables - The group looked back at the specifications that we enumerated on the previous deliverables - We will end up using all of our EERDs - Importance of two types of logins (Administrator View and Student View) - Administrator can add new fields, view all - Students will fill up, and can only see their own	- Create summary list of deliverables

ACTION ITEMS		
Action Item	**Point Person**	**Due Date**
- Discussion of Implementation	Wesley and Vonn	April 28
- Final Logical ERD	Wesley	Before May 14
- MySQL Commands	Vonn and Andre	May 1
- Front-End – Draft	Ken and Nigel	April 28
- Connecting Front End and Back-End	Ken and Mik	April 30
- Google Drive for documenting project	Mik	April 26
- Fake Data	Mik and Ken	April 30

NEXT MEETING					
Date	April 30, 2018	Time	7:30pm	Location	Online
Agenda	Updates on action items				

MEETING DOCUMENTATION: Project Deliverable 3

MEETING DETAILS			
Date	05-12-18	Start Time	09:00
Location	Online	End Time	10:30
Documented by	Alog, Kenneth		

AGENDA

- Updates on action items
- Finalization of project deliverable
- Final debugging of project code

ATTENDANCE (P=Present / A=Absent / L=Late Arrival / E=Early Departure)			
P	Acosta, Vonn	P	Ong, Wesley
P	Alog, Kenneth	P	Rondolo, Andre
P	Fuentes, Mik	P	Yu, Nigel

DISCUSSIONS

Topic / Discussion Point	Action Items
- Debugging of SQL codes and connecting it to front-end - Test out the SQL commands with dummy student data - Discuss limitations of the enlistment system - Finalization of the project deliverable 3 by May 14	- Debug SQL commands - Provide dummy data to test and test it - Finalize deliverable

ACTION ITEMS

Action Item	Point Person	Due Date
- Debug SQL commands; provide dummy data to test and test it	Ken, Vonn, Andre and Mik	May 14
- Software and hardware requirements	Wesley and Nigel	May 14
- Limitations of the System	Nigel and Ken	May 14
- Tables and Integrity Constraints	Vonn and Andre	May 14
- Sample Queries	Vonn and Andre	May 14
- Final Logical ERD	Wesley	May 14
- Final Set of Screens and Reports	Mik	May 14

- Meeting Documentation Compilations	Wesley	May 14

NEXT MEETING					
Date	N/A	Time	N/A	Location	N/A
Agenda	N/A				