TABLE OF CONTENTS

[CHAPTER 7: PROJECT EVALUATION 2](#_Toc10721761)

[1.0 INTRODUCTION 2](#_Toc10721762)

[1.1 ACHIEVEMENTS 2](#_Toc10721763)

[1.2 LIMITATIONS 2](#_Toc10721764)

[1.3 RECOMMENDATIONS 2](#_Toc10721765)

[1.4 CONCLUSION 3](#_Toc10721766)

[2 BIBLIOGRAPHY 4](#_Toc10721767)

[2.0 REFRENCES 4](#_Toc10721768)

[2.1 APPENDIX A: USER MANUAL 6](#_Toc10721769)

[2.1.1 Installation 6](#_Toc10721770)

[2.2 APPENDIX B: DOCUMENTS STUDY GUIDE 8](#_Toc10721771)

[2.3 APPENDIX C: INTERVIEW 8](#_Toc10721772)

[2.4 APPENDIX D: QUESTIONNAIRE 8](#_Toc10721773)

[2.5 APPENDIX E: CODE SNIPPET 10](#_Toc10721774)

[2.5.1 Login Provider 10](#_Toc10721775)

[2.5.2 Upload Applicant Files 12](#_Toc10721776)

[2.5.3 Helper Functions 13](#_Toc10721777)

[2.5.4 Login UI 15](#_Toc10721778)

[2.5.5 Backend API routes 16](#_Toc10721779)

[2.5.6 Backend Generate Timetable function 17](#_Toc10721780)

# CHAPTER 7: PROJECT EVALUATION

## INTRODUCTION

Project evaluation is a systematic and objective assessment of an ongoing or completed project. The aim is to determine the relevance and level of achievement of project objectives, development effectiveness, efficiency, impact, and sustainability. Evaluations also feed lessons learned into the decision-making process of the project stakeholders, including students and all the beneficiaries of the project. Evaluation is also an important part of this project as it also shows the recommendations and the next steps.

## ACHIEVEMENTS

The screen dumbs inserted in the preceding chapter clearly shows that in relation to the objectives stetted in the proposal the developer achieved managed to meet the objectives. Other achievements include:

* Deploy the app onto the Android mobile device and successfully run
* The major functionality of the application measured against the current portal is working
* Notification and alerts capability
* Login, remember password and re-authenticate after a certain period of time functionalities are working
* Admission via the application is working with acceptance notifications
* Emergency, news, and calendar are now operational.

## LIMITATIONS

Developing software is a lifetime project as long as the users are still using it. Things like bugs fixes, functionality improvements, and additional functionality always suggested by users hence the major limitation is that they are other functions that are ignored on the first build but will be implemented in the coming version.

Another limitation is finance. There are APIs and libraries that would have made the project fast, efficient and easy but they require licenses and purchases which the developer is short of.

Other limitations include:

* Limited testers and developers to make the development process fast.
* Limited memory and space for implementing fast systems
* Rights and permissions to use the actual backend platform

## RECOMMENDATIONS

Technology advance every single day, in order to keep up with it, systems should be kept up-to-date, as a recommendation, it is important to have a development team dedicated on monitoring how the developed system functions, constantly checking bugs and users queries through social media.

After application accepted, the default student password is their phone number which they used to log in, and they will get a message, it is important to change it to a secure password as soon as they logged in for the first time.

Users, which are students should keep their login credentials safe to protect their confidential information because anyone who can access them can literally able to log in on any device and see every academic and personal information.

## CONCLUSION

As dedicated as the developer was, the proposed system came out great and the developer learned a lot along the way. Project management, time management, research, and dedication are skills that the developer gained in developing this system. The last section is the user manual to guide the users on how to use the system.

# BIBLIOGRAPHY

## REFRENCES

Barry, C., Conboy, K., Lang, M., Wojtkowski, G. and Wojtkowski, W. (2008). *Information Systems Development*. Dordrecht: Springer.

Bourgeois, D. and Bourgeois, D. (2019). *Chapter 10: Information Systems Development*. [online] Bus206.pressbooks.com. Available at: https://bus206.pressbooks.com/chapter/chapter-10-information-systems-development/ [Accessed 9 May 2019].

Dennis, A., Roth, R. and Wixom, B. (2015). *Systems analysis and design*. Hoboken, NJ: Wiley.

Dennis, A., Wixom, B., Tegarden, D. and Seeman, E. (n.d.). *System analysis & design*.

Hanby, J. (2019). *Software Maintenance: Understanding and Estimating Costs*. [online] LookFar. Available at: http://blog.lookfar.com/blog/2016/10/21/software-maintenance-understanding-and-estimating-costs/ [Accessed 10 May 2019].

Holzinger, A. (2000). *Multimedia basics*. New Delhi: Firewall Media.

Hsu, C. C. and Sandford, B. A. (2007). *The Delphi Technique: Making Sense of Consensus*, *Practical Assessment, Research & Evaluation, Vol 12, No 10 3*.

J., M. (2011). Economic and Operational Feasibility Analysis of Solid Waste Minimization Projects. *Integrated Waste Management - Volume I*.

James, P. K. (1997). *Research Design in Occupational Education*, Oklahoma State University, United States of America.

Jawadekar, W. S. (2004). *Software Engineering: Principles and Practice:*  *Tata McGraw Hill India.*

Jonasson, H. (2013). *Determining Project Requirements* (2nd Ed), Boca Raton Taylor and Francis Group, LLC. United States of America.

Kendall S, Kenneth E, Julie E. (2014). *Systems Analysis and Design*, Englewood Cliffs, NJ: Prentice Hall.

Khan, F. and Khan, M.E. (2012*). International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 3, No.6.*

Kahate, A. (2004). *Object Oriented Analysis & Design*. New Dehli: Tata McGraw-Hill Education.

MacConnell, S. (1996). *Rapid development*. 1st ed. Redmon: Microsoft Press.

Marchewka, J. (2015). *INFORMATION TECHNOLOGY PROJECT MANAGEMENT 5E*. New York: Wiley.

Pham, H. (2000). *Software reliability*. Singapore: Springer.

Rivero, L., Doorn, J. and Ferraggine, V. (2006). *Encyclopedia of database technologies and applications*. Hershey, PA: Idea Group Reference.

Roth, R., Dennis, A. and Wixom, B. (2013). *System analysis and design*. Hoboken: J. Wiley & Sons.

Saleh, K. (2009). *Software engineering*. Fort Lauderdale: J. Ross.

Sommerville, I. (2008). *Software engineering*. Addison Wesley.

Tilley, S. and Rosenblatt, H. (n.d.). *Systems analysis and design*. 10th ed.

## APPENDIX A: USER MANUAL

### Installation

Currently, the application can be downloaded from a third party web site <https://creative.joemags.co.zw>, intended for early adopters for testing and bugs reports. And only the developer has compiled for android but soon it will be on both IOS and Android and will be downloaded from official respective app stores.

**NOTE:** Allow installation from unknown sources in your Android settings.

|  |  |  |
| --- | --- | --- |
| C:\Users\jmuna\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Download Andr.png | C:\Users\jmuna\AppData\Local\Microsoft\Windows\INetCache\Content.Word\clickOk.png | C:\Users\jmuna\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Downloading.png |

Download URL

Click the download button on the GZU Student Portal App and accept the installation.

|  |  |  |
| --- | --- | --- |
|  | C:\Users\jmuna\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Unknown Sources.png | C:\Users\jmuna\AppData\Local\Microsoft\Windows\INetCache\Content.Word\App Installed.png |

Install the application

|  |  |  |
| --- | --- | --- |
|  |  |  |

The application should run successfully.

## APPENDIX B: DOCUMENTS STUDY GUIDE

The following documents were very helpful to the analyst in the design and development of the application:

* GREAT ZIMBABWE UNIVERSITY STUDENT PORTAL fees pdf file accessed from <http://www.gzu.ac.zw/wp-content/uploads/2016/08/FEE-STRUCTURE-2019.pdf>
* UNDERGRADUATE and PostGrad ADMISSION APPLICATION form <http://www.gzu.ac.zw/wp-content/uploads/2019/05/UNDERGRADUATE_ADMISSION_APPLICATION_CURRENT.pdf>
* Documents that can be accessed from <http://www.gzu.ac.zw/downloads/>

## APPENDIX C: INTERVIEW

The analyst conduct interviews and below are the questions used for the structured interviews:

* Interviewee Name
* Program and Academic Year
* Technological gadgets they own and use
* Type of mobile phones and their operating system
* If they know and remember the student portal URL
* How often they log in to their student portal
* How easy or difficult it is to use the online student portal
* If they or they don’t use any mobile application and how often they install a new application
* Average applications they use

## APPENDIX D: QUESTIONNAIRE

Together with Interviews, the analyst also sent out questionnaires offline using printed papers and online using Survey Monkey and Google forms. The following questions were in the questionnaires.

**Instructions**: Tick/Check where applicable.

1. Write your Registration Number……………………………………...
2. What is the type of your mobile phone?
   * Android
   * IOS
   * Window
3. How often do you access your student portal?
   * Multiple times a week
   * Once a week
   * Once a month
   * On registration and when exams are out only
4. What challenges do you face when accessing your student portal? ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………
5. Do you use any mobile applications? If yes, state one ……………………………………….
6. Do you install any applications on your mobile application? If yes, what is the source of the applications? (You can tick multiple)
   * App stores
   * Share It
   * Amazon
   * Other. Mention………………………………………
7. Would you say Mobile applications are better than web sites?
   * Yes
   * No
   * Maybe
8. If Yes above, in what way do you think they are better? ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………
9. How likely are you to use GZU mobile application than website portal if it is available today?
   * Very Likely
   * Likely
   * Neutral
   * Not Likely
   * Very Unlikely

***Thank you for your time.***

## APPENDIX E: CODE SNIPPET

### Login Provider

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { BehaviorSubject, Subject } from 'rxjs';

import { Storage } from '@ionic/storage';

import Swal from 'sweetalert2'

import 'rxjs/add/operator/timeout';

import { Events } from 'ionic-angular';

import { timer } from 'rxjs/observable/timer';

import { CacheService } from 'ionic-cache';

import { getEnv } from '../../common/is-cordova-available';

@Injectable()

export class AuthProvider {

public authState: Subject<any> = new BehaviorSubject(undefined);

public nextStep: Subject<any> = new BehaviorSubject(null);

baseUrl: string = "";

public user: any;

constructor(public *http*: HttpClient, public *storage*: Storage,

private *cache*: CacheService,

private *event*: Events) {

this.baseUrl = getEnv().BASE\_URL + '/api/auth';

console.log(this.baseUrl);

this.storage.get('USER').then(*data* => {

this.authState.next(data);

})

this.authState.subscribe(*user* => {

if (user) {

this.user = user;

}

})

console.log('Hello AuthProvider Provider');

}

public logout() {

this.storage.remove('USER').then(() => this.authState.next(null));

}

public login(*data*: any) {

return this.http.post(`${this.baseUrl}/login`, data).timeout(15000);

}

public userDetails(*token*: string) {

let headers = new HttpHeaders()

.set('Content-Type', 'application/json')

.set('Authorization', 'Bearer ' + token);

return this.http.post(`${this.baseUrl}/me`, {}, { headers }).timeout(15000);

}

public enterPassword(*user*, *where* = "global") {

let { reg\_number } = user;

Swal.fire({

title: 'Login expired. Please enter your password',

input: 'password',

inputAttributes: {

autocapitalize: 'off'

},

showCancelButton: true,

confirmButtonText: 'Login',

showLoaderOnConfirm: true,

preConfirm: (*password*) => {

let cred = {

password,

reg\_number

}

return fetch(this.baseUrl + '/login', {

method: "POST",

headers: {

"Content-Type": "application/json",

},

body: JSON.stringify(cred)

}).then(*response* => {

if (!response.ok) {

throw new Error(response.statusText)

}

return response.json()

}).catch(*error* => {

Swal.showValidationMessage(

`Login failed, Invalid credentials`

)

})

},

allowOutsideClick: () => !Swal.isLoading()

}).then((*result*) => {

if (result.value) {

user.access\_token = result.value.access\_token;

this.storage.set('USER', user).then(() => {

this.authState.next(user);

this.event.publish(`login:${where}`, result.value.access\_token);

this.event.publish(`TOKEN:changed`, result.value.access\_token);

});

*//this.nextStep.next(result.value.access\_token);*

}

})

}

getProfile(*token*: string) {

let headers = new HttpHeaders()

.set('Content-Type', 'application/json')

.set('Authorization', 'Bearer ' + token);

const URL = `${this.baseUrl}/profile`;

const req = this.http.post(URL, {}, { headers });

return this.cache.loadFromObservable(URL, req);

}

}

### Upload Applicant Files

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Injectable } from '@angular/core';

import 'rxjs/add/operator/timeout';

import { getEnv } from '../../common/is-cordova-available';

@Injectable()

export class FileHandlerProvider {

url: string = '';

constructor(public *http*: HttpClient) {

this.url = getEnv().BASE\_URL + '/api/auth/filehandler';

console.log(this.url);

}

postFile(*file*: Blob, *name*) {

let httpOptions = {

headers: new HttpHeaders({

'enctype': 'multipart/form-data; boundary=----WebKitFormBoundaryuL67FWkv1CA'

})

};

let formData = new FormData();

formData.append('file', file, name);

return this.http.post(this.url, formData, httpOptions).timeout(15000);

}

delFile(*fileName*: string) {

let url = `${this.url}?fileName=${fileName}`;

return this.http.delete(url);

}

}

### Helper Functions

import { Injectable } from '@angular/core';

import { ToastController, LoadingController } from 'ionic-angular';

@Injectable()

export class GlobalFuncsProvider {

\_toaster: any;

\_loader: any;

public loaded: boolean = false;

constructor(

private *toastCtrl*: ToastController,

private *loadingCtrl*: LoadingController

) { }

public showToster(*message*, *type*: string = 'info-toaster', *duration*: number = 3000) {

if (this.\_toaster && type !== 'info-toaster') {

this.\_toaster.dismiss();

}

let position = (type === 'success-toaster') ? 'top' : 'bottom';

this.\_toaster = this.toastCtrl.create({

message,

duration,

position,

cssClass: type,

dismissOnPageChange: true

});

this.\_toaster.present();

this.\_toaster.onDidDismiss(() => {

this.\_toaster = null;

})

}

public showLoading(*content*: string = 'Loading..', *duration*: number = 3000, *callback*: any = (() => console.log('Dissmised'))) {

if (this.\_loader) {

this.\_loader.dismiss();

}

this.\_loader = this.loadingCtrl.create({

spinner: 'crescent',

content: content,

duration

});

this.\_loader.present();

try {

this.\_loader.onDidDismiss(() => {

callback();

if (!this.loaded) {

this.showToster('Network error, try again!', 'error-toaster');

this.loaded = true;

}

});

} catch (error) {

}

}

public destroyLoader() {

if (this.\_loader) {

this.\_loader.dismiss();

this.\_loader = null;

}

}

}

### Login UI

<ion-content no-padding class="clearfix pt4 splash-grad">

<div class="sm-col-6 md-col-3 lg-col-4 mx-auto">

<div class="wrap-login100">

<form class="login100-form validate-form" [formGroup]="loginForm" (ngSubmit)="submit(loginForm.value)">

<span class="login100-form-logo">

<img src="../../../assets/imgs/portal.png">

</span>

<span class="login100-form-title p-b-34 p-t-27">

Student Portal

</span>

<div class="wrap-input100 validate-input {{(reg\_number\_inv? 'animated infinite shake': '')}}">

<input #regNumber formControlName="reg\_number" (focus)="(focused = true);" (focusout)="focused = false"

style="text-transform:capitalize" class="input100" type="text" name="reg\_number" placeholder="Reg Number"

required>

<span class="focus-input100" data-placeholder="&#xf207;"></span>

</div>

<div class="wrap-input100 pass-input validate-input {{(password\_inv? 'animated infinite shake': '')}}">

<input #pass formControlName="password" (focus)="(focused = true);" (focusout)="focused = false"

class="input100" type="password" name="password" placeholder="Password" required>

<span class="focus-input100" data-placeholder="&#xf191;"></span>

</div>

<div class="container-login100-form-btn p-t-27">

<button ion-button block class="login100-form-btn" color="light">

LOGIN

</button>

</div>

<p text-center padding-top class="looking\_txt">ADMMISSION OPEN FOR AUG 2019</p>

</form>

</div>

<div padding-right padding-left>

<button ion-button outline block color="secondary" (click)="goToApply()">

APPLY

</button>

</div>

</div>

</ion-content>

### Backend API routes

<?php

use Illuminate\Http\Request;

*/\**

*|--------------------------------------------------------------------------*

*| API Routes*

*|--------------------------------------------------------------------------*

*|*

*| Here is where you can register API routes for your application. These*

*| routes are loaded by the RouteServiceProvider within a group which*

*| is assigned the "api" middleware group. Enjoy building your API!*

*|*

*\*/*

Route::fallback(function() {

return response()->json(['error' => 'Resource not found'], 404);

})->name('fallback');

Route::group(['prefix' => 'auth'], function ($router) {

Route::post('login', 'StudentController@login');

Route::post('logout', 'StudentController@logout');

Route::post('refresh', 'StudentController@refresh');

Route::post('me', 'StudentController@me');

Route::post('profile', 'StudentController@getProfile');

Route::post('apply', 'API\ApplicantsController@apply');

Route::get('programmes', 'API\ApplicantsController@programmes');

Route::post('filehandler', 'API\ApplicantsController@filehandler');

Route::delete('filehandler', 'API\ApplicantsController@delFile');

});

Route::group(['prefix' => 'app'], function ($router) {

Route::post('emergency', 'API\EmergencyController@store');

Route::get('financials', 'API\MobileApp@getFinancials');

Route::get('transactions', 'API\MobileApp@getTransactions');

Route::get('courses', 'API\MobileApp@getCarriculumCourses');

Route::get('timetable', 'API\MobileApp@getTimetable');

Route::post('store\_deposit', 'API\MobileApp@storeDeposit');

Route::post('register', 'API\MobileApp@storeRegisteredCourses');

});

### Backend Generate Timetable function

public function generateTimetable(){

$dates = request()->dates;

if(!is\_array($dates)){

return response()->json(['error'=>true, 'message' => 'Invalid range, try again'],400);

}

$dates\_collection = collect($dates);

$reg\_courses = RegisteredStudent::all();

$venues = Venue::all();

if(count($venues) === 0){

return response()->json('add\_venue', 400);

}

*//TODO: Auto generate timetable base on at leas 1 registered student.*

$group\_by\_course = unique\_multidim\_array($reg\_courses,'course\_id');

*//Group by courses*

$unfinised = false;

foreach ($group\_by\_course as $registered) {

$group = RegisteredStudent::whereCourseId($registered['course\_id'])->get();

*//check level*

if(count($group) != 0){

$first\_student = AcademicInfo::whereRegNumber($group[0]['reg\_number'])->first();

$level = $first\_student->level;

$allowed\_venues = Venue::where('capacity', '>=', count($group) + 5)->get();

*//Filter venues that can accommodate group either morning or evening*

*// $allowed\_venues = $allowed\_venues->filter(function ($venue, $key) use ($group) {*

*// return ($venue['capacity'] - (count($group) + 5)) > 0; //Space for 10 late registration*

*// });*

*//FIXME: Create a function for the for eaches below and if fail rerun it with getDateRange($dates\_collection, $level, true)*

$all\_dates = $this->getDateRange($dates\_collection, $level);

$selected\_venue = null;

$writting\_date = null;

foreach ($all\_dates as $date) {

*#testing venue against time;*

$\_student\_writing = false;

foreach ($group as $\_student) {

$no\_student\_writting = RegisteredStudent::whereRegNumber($\_student['reg\_number'])->whereExamDate($date)->first();

if(isset($no\_student\_writting)){

$\_student\_writing = true;

break 1;

}

}

if(!$\_student\_writing){

foreach ($allowed\_venues as $vnu) {

$test\_vacancy = VenueTimetable::whereVenueId($vnu->id)->whereExamDate($date)->get();

if(count($test\_vacancy) === 0){

$selected\_venue = $vnu;

break 1;

}

$group\_size = $test\_vacancy->sum('group\_size');

if($vnu->capacity > $group\_size && ($vnu->capacity - $group\_size) > (count($group) + 5)){

$selected\_venue = $vnu;

break 1;

}

}

if(isset($selected\_venue)){

$writing\_date = $date;

break 1;

}

}

}

if(!isset($writing\_date) || !isset($selected\_venue)){

return response()->json("add\_venue", 400);

}

$v\_ttable = VenueTimetable::firstOrNew(

array('venue\_id' => $selected\_venue->id,

'course\_id' => $registered['course\_id']));

$v\_ttable->group\_size = count($group) + 5;

$v\_ttable->exam\_date = $writing\_date;

$v\_ttable->save();

foreach ($group as $registered\_stu) {

RegisteredStudent::where('id', $registered\_stu->id)->update(['exam\_date' => $writing\_date]);

}

}

}

if($unfinised){

return response()->json('venue\_unfit', 400);

}

$ttable = AppSettings::firstOrNew(array('name' => 'exam\_timetable'));

$ttable->status = true;

$ttable->save();

OneSignal::sendNotificationUsingTags(

"Examination timetable is out",

array(

["field" => "tag", "relation" => "=","key" => "exam\_timetable", "value" => "1"],

),

$data = null,

$buttons = null

);

return response()->json('generated' , 200);

}

function getDateRange($dates, $level, $nochunk = false){

$chunks = $dates->chunk(5)->toArray();

$find\_dates = [];

if($level == '1.1' || $level == '1.2'){

$find\_dates = $chunks[2];

}else if($level == '2.1' || $level == '2.2'){

$find\_dates = $chunks[1];

}else if($level == '4.1' || $level == '4.2'){

$find\_dates = $chunks[0];

}

if($nochunk){

$find\_dates = array\_reverse($dates->toArray());

}

$all\_dates = [];

foreach ($find\_dates as $d) {

$time = ['09:00:00', '14:00:00'];

for ($i=0; $i < 2; $i++) {

$all\_dates[] = $d.' '.$time[$i];

}

}

return $all\_dates;

}