|  |  |
| --- | --- |
|  | **Qatar University**  **College of Engineering**  **Department of Computer Science and Engineering** |

Senior Project Report

**ParQU - Parking System Using Cloud Computing Based on IoT**

**Project Group Members:**

Amal Khaled Haidarah (201302862)

Alaa Mousa Mousa (201305148)  
Ghareisa Mohammed Al-Kuwari (201402464)

**Supervisor**: Dr. Abdulaziz Al-Ali

**2019**

This project report is submitted to the Department of Computer Science and Engineering of Qatar University in partial fulfillment of the requirements of the Senior Project course.

# Declaration

This report has not been submitted for any other degree at this or any other University. It is solely the work of us except where cited in the text or the Acknowledgements page. It describes work carried out by us for the capstone design project. We are aware of the university’s policy on plagiarism and the associated penalties and we declare that this report is the product of our own work.

Student: Date:

Signature:

Student: Date:

Signature:

Student: Date:

Signature:

# Abstract

< ToDo: The abstract is a brief overview of your project and its objectives. It should present an accurate summary of the problem your project has addressed and a summary of your solution. It should provide a concise and a clear snapshot of your project to the reader. The length of your abstract should not exceed 500 words.

/>

<ToDo – **Writing the final report**

To produce the final report, you can follow the following recommended steps:

* Where appropriate, copy material from your interim report into the final report. Go through that material, and update it based on changes that have occurred in your project between last semester and now.
* Revise the Abstract and enhance it by adding the project’s key achievements and most important conclusions. The last paragraph should highlight the novelty of your design (e.g., what makes your design unique and what are the impacts of your engineered solution, etc.).
* Fill in all of the appropriate material required for the final report.
* Update the Table of Content, the List of Figures and the List of Tables.
* Review the whole document to make sure that it is coherent and to ensure that it addresses all the requirements listed in the Project Guide and the Project Grading Rubrics. Also make sure that the tense used is the present tense and the past and not the future (e.g., avoid ‘we will’ or ‘system should’ and report what has been done).
* Seek your supervisor’s feedback and address any issues raised.
* Note that the template is only provided as a guide. In consultation with your supervisor, you can add other sections to align it with the nature of your project.
* If you are using Word ‘Track Changes’ you must accept all the changes before submitting the softcopy of your report.

/>

# Acknowledgment

< ToDo: Acknowledge any assistance you received for your project. />

Table of Contents

[Declaration ii](#_Toc1152088)

[Abstract iii](#_Toc1152089)

[Acknowledgment iv](#_Toc1152090)

[List of Figures vi](#_Toc1152091)

[List of tables vi](#_Toc1152092)

[1. Introduction and Motivation 7](#_Toc1152093)

[2. Background and related work 8](#_Toc1152094)

[2.1. Background 8](#_Toc1152095)

[2.2. Related work 8](#_Toc1152096)

[3. Requirements analysis 9](#_Toc1152097)

[3.1. Functional requirements 9](#_Toc1152098)

[3.2. Design constraints 9](#_Toc1152099)

[3.3. Design standards 10](#_Toc1152100)

[3.4. Professional Code of Ethics 10](#_Toc1152101)

[3.5. Assumptions 10](#_Toc1152102)

[4. Solution design 11](#_Toc1152103)

[4.1. Overview 11](#_Toc1152104)

[4.2. Hardware/software used 11](#_Toc1152105)

[4.3. Hardware design 11](#_Toc1152106)

[4.4. Software design (Structural model => class diagram)/ (behavior model (Activity diagram + sequence diagram + state diagram) / (database design) / (user interface design) / ( design patterns) 12](#_Toc1152107)

[5. Implementation 13](#_Toc1152108)

[6. Testing 14](#_Toc1152109)

[7. Evaluation of the impact of the engineered solution 15](#_Toc1152110)

[8. Conclusion 16](#_Toc1152111)

[9. Future work 17](#_Toc1152112)

[10. Student reflections 18](#_Toc1152113)

[References 19](#_Toc1152114)

[Appendix A – Project Plan 20](#_Toc1152115)

[A.1. Project milestones 20](#_Toc1152116)

[A.2. Project timeline 20](#_Toc1152117)

[A.3. Anticipated risks 20](#_Toc1152118)

[Other Appendices 21](#_Toc1152119)

# List of Figures

# List of Tables

# Introduction and Motivation

## Problem statement

## Project significance

## Project objectives

# Background and related work

## Background

< ToDo: Discuss briefly the major concepts, issues and key problems related to your project. This should give the reader the necessary background information to understand your project. />

## Related work

<ToDo: Discuss the related work and how others researchers or developers have dealt with the problem at hand. Cite appropriate references including relevant research papers. A list of 5 to 10 references should be adequate to include in this section. Seek your supervisor assistance to locate relevant related work.

At the end of this section, you should highlight how your project is different than previous approaches. Discuss how similar and/or different your project is expected to be compared to the existing designs using various features. These features may include aspects of the designs such as input modes (various types of inputs), processor modes (single/multiple, number of cores, actuators, etc…). You may use a table to summarize the key similarities and differences between your project and the related work. />

# Requirements analysis

< ToDo: In this section you are presenting the requirements that your solution to the problem needs to meet. />

## Functional requirements

< ToDo: Collect, document and analyze the project functional requirements. The requirements specification should be done using suitable methods and tools. You may include use cases if they are relevant for your project. Refer to your software engineering course or to posted slides for further details about use cases.

The Functional Requirement Specifications document the operations and activities that a system must be able to perform. They should include:

* Description of data/signal to be entered into the system
* Description of operations/processes performed by each functional section of the system
* Description of work-flows performed by the system
* Description of system reports/actuations or other outputs
* Who are the main users of the system

The functional specification is designed to be read by a general audience. Readers should understand the system, but no particular technical knowledge should be required to understand the functional requirement specifications.

/>

## Design constraints

< ToDo:

* With the help of your supervisor, list and explain the constraints that the proposed solution must adhere to, such as technical, economic, environmental, social, ethical, health and safety.
* **Based on the examples provided on ‘Design Constraints and Design Standards Guide’ and with the help of your supervisor**, make lists of your **technical** and **practical** design constraints. Make sure that the identified constraints are relevant and applicable to your project. Your design and implementation will need to meet the identified constraints.
* You also need to justify and explain *why* you need each constraint, and discuss how each is relevant to your project.
* Desired quality attributes should be *specific* and *measurable*. Instead of just saying the system should be “fast” or “fast enough to process the data”. You need to give specific time constraints. For example, “The system should respond to a search quickly” is a bad requirement. While it is specific (it means response time to searches) it isn’t measurable (you can’t measure ‘quickly’.) You should be more specific and say something like “The system should return search results in less than 200 ms, 95% of the time”. Instead of just stating that the system should be secure, you should specify what accidents and attacks must be prevented, what types of vulnerabilities the system must not incorporate and what hazards and threats it must defend against.
* Think of these constraints as boundaries to the above Functional Requirements and hence defines scope of your design. Select 3 to 5 realistic constraints since selecting too many will cause more problems in the design work in case of issues arising related to components availability, and costs, etc… In addition, constraint selection should be done bearing in mind the fact that your solution design should be done with respect to these constraints. Additionally during testing phase you should prove that your solution meets the identified constraints.

/>

## Design standards

<ToDo:

With the help of your supervisor, list the design standards that the proposed solution need to comply with. **For further details refer to the ‘Design Constraints and Design Standards Guide’.**

/>

## Professional Code of Ethics

<ToDo:

Identify the legal, security, and social issues and responsibilities relevant for your project. Then from ACM and the IEEE Software Engineering Code of Ethics and Professional Practice select the most relevant items to address the identified issues. You need to discuss and justify using/practicing them during the course of the project work. The selection should be based on appropriate project functionalities, development tasks, general conduct of the team and interactions with other project stakeholders.

**Do not cut and paste the entire ACM and IEEE codes of ethics into your report**. Instead, read through them and in your report specifically address the ones most relevant to your project and discuss how you are considering them in your project.

/>

## Assumptions

<ToDo:

List any specific project assumptions. An assumption is something that we cannot establish as being true at this point in time, but is likely to be true. For example, if you are building a system for analysis of Doha traffic you can assume that you will get the traffic data on time from the Traffic Authority. Assumptions are potential failure points in a project. They need to be monitored and managed. A bulleted list of items is the best format to use for this section. Again, be realistic in making these assumptions since some of these may be false from the beginning. Consult with your supervisor for such selections.

/>

# Solution design

<ToDo:

In this section, you need to document the detailed design specifications of the hardware and software subsystems used in your design to meet the functional requirements and the practical design constraints of your project. You should document your detailed design using suitable methods and tools such as circuit diagrams, logic diagrams, block diagrams, flow charts, class diagrams and sequence diagrams. Also you need to provide detailed justification of design choices. Additionally, you should highlight the novel aspects of your design. Additionally, you should also **evaluate the effect of design choices** on your system quality attributes**.**

You could consider breaking this section into the sub-sections listed below. Note that you are *not* required to include all of these sub-sections, instead you need to decide how to best communicate your solution’s design.

/>

## Overview

< ToDo:

* Present an overview of the proposed solution. This can be done in the form of a simple one or two paragraphs giving a birds-eye view of the solution. The description should be kept very generic so that even a layman will be able to comprehend the scope of your work.
* Add a high level architecture diagram of the proposed solution. The diagram should show how your solutions is decomposed and organized into components.
  + A recommended way of presenting this is through a Block Diagram or Illustrative Block diagrams in which some components/blocks can be illustrations/images of the actual component.
  + Describe the role and the interfaces of key components of your high level architecture.
  + Discuss the key interactions between the identified components.
  + While one Block diagram can be used to show hardware components/connectivity, another Block diagram or Flow chart can be used to do the same for the software aspect of the project. You should present the software logic at this point as a high level abstraction.

/>

## Hardware/software used

< ToDo: Discuss the hardware/software platforms and tools that were used for the design and implementation and justify your choices. Also list the hardware/software platform(s) on which the system runs. />

## Hardware design

<ToDo:

For major hardware subsystems in your design, you must present the theory behind the different technological approaches, the tradeoffs associated with each approach, and your justification for selecting a particular approach. For instance, selection of a particular microcontroller for your design; what were the main factors that contributed to this specific choice. For example, you may choose between Arduino, MC68HCxxx, BasicStamp, and PIC18Fxx and you have decided to use PIC18F47 due to:

1. its programming compatibility with C
2. its low cost
3. its development platform availability
4. ability to input analog signals…

To document the design of the hardware components you can use as many of the following as possible:

* Circuit Diagram (Showing actual circuit diagram with all the components used)
* Connectivity diagram for modular hardware.
* Logic Diagrams (Showing logic flow with respect to signals/numbers)
* Functional Diagrams (showing subsets of the circuit as functional blocks)
* State Diagrams (to explain the sequential logic flow)
* Any other drawings to communicate your design

Organize the content of this section using appropriate subsections. It is recommended to have a sub-section of each of the recommended artifacts listed above.

/>

## Software design (Structural model => class diagram)/ (behavior model (Activity diagram + sequence diagram + state diagram) / (database design) / (user interface design) / ( design patterns)

<ToDo:

You should document the structural and the behavioral aspects of your software components.

For projects that only use embedded code (e.g., programming microprocessor), a flow-chart describing your software design could be sufficient. Projects with user interfaces may elaborate on the functioning of the interface using some of the models mentioned below.

The software components could be documented using:

* Class diagram for the whole system. While Class may be too specific to JAVA or C++ environments, you may use equivalent components applicable to your design. For instance, VI for LabVIEW designs, MDL files in MATLAB, etc… If the model is too big, partition the diagram using some reasonable criteria. For example, you may provide the entity classes and the controller classes as separate diagrams.
* Specific emphasis should be given to the elaboration of the software components that are responsible for interfacing with the hardware components of your design. For example, if a protocol-like procedure was developed in connecting to a hardware module then it should be explained in detail. Such components may include packetization/depaketization procedures, etc…
* Wherever applicable, all associations between classes/software-modules should be identified through defining the association name and the multiplicities on both ends. Aggregation and inheritance relationships should be identified. A brief explanation should accompany each diagram.
* Overall software logic may also be described in order to know the appropriate logic flow. In case of LabVIEW or SIMULINK programs, this may not be needed. For other programming environments, the state diagram or extended flow chart may be sufficient.
* In case of graphical codes (such as LabVIEW programs) the software structure may be described by various sub-Vis of the system. Similarly for SIMULINK diagrams, the models should be explained individually. The complete program (graphical/text) should be included in appendix and should not be listed completely in this section.

/>

# Implementation

<ToDo:/>

* Actual Design description with pictures and diagrams. E.g., a “wiring diagram” of the implemented hardware can be added.
* Actual images of various modules must be included wherever possible. Otherwise, at least the images of various aspects of the completed design must be shown.
* List the different tools and framework used for the implementation.
* Discuss any novel aspects of your implementation (if applicable). You may link this aspect of your design to the comparison table at the end of literature review and elaborate on the steps taken in achieving these novelties in your design.
* Discuss the challenges encountered during the implementation and how they were addressed.

You may organize any of the above recommended points as subsections.

/>

# Testing

<ToDo:

In this section you should **describe in detail the tests** you have conducted to verify that your prototype satisfies the desired functional requirements while meeting the design constraints. The testing should verify and provide evidence that your solution solved the stated problem and satisfied the requirement specifications (if not explain what is lacking). This will also allow you to find defects then fix them and also identify possible improvements.

* Describe in detail the tests you have run to verify that your prototype satisfies each of the functional requirements of your project. For example, if you used Use Cases to describe your non-functional requirements, then for each use case you should write a test case, run it and report the testing results (the key test cases can be added as an Appendix). Functional testing will allow you to find defects then fix them and also identify possible improvements. You need to have a comprehensive set of tests that verifies correct functionality of every component of your system.
* You should conduct appropriate tests to verify that the prototype meets the design constraints, documented in Section 3.2 of the interim report. This section must present sufficient evidence and clear discussion that your design and prototype **met/did-not-meet all of the identified technical and practical constraints** documented in Section 3.2. You may use a summary table to elaborate these points. The table columns could contain:
* Brief description of the constraints
* Explanation of the testing steps taken to evaluate if the constraint was met or not.
* Measurements data that prove your system met/did-not-meet the constraint. Also compare measured data against expected values, then include a *%error* as (actual – expected)/expected\*100%, and use only three digits of precision. In case a constraint is not met, then explain the reason for that. .

For further details and examples refer to the document titled ‘*Examples of how to address and verify some of the design constraints*’ posted on the Senior Projects website.

* Describe in detail the tests performed on the individual subsystems of your project. Discuss how you tested each subsystem and what the results of each test were.
* In addition, do not forget to include:
* If you have a GUI of some type, you need a screen shot of it.
* If you have a physical display of some type (LEDs, LCDs, etc.), you need a photograph of the display showing typical operation.
* For any interfaces of your hardware components – USB, I2C, SPI, RS232, parallel interfaces, or A/D inputs – you may show oscilloscope pictures that demonstrate a sample data transfer of this interface and the typical voltage/frequency ranges.

You should **present the test results**, with appropriate level of details in addition to accuracy and completeness**,** using tables, graphs, diagrams, screen shots etc. Additionally, **discuss these results** and explain whether the prototype has achieved the requirements. If not state what is lacking or still need improvement then explain the reason for that. There will probably be multiple subsections under this section to describe each system test and its result.

Note that this section is a substantial portion of the grade for your final report, and will require a significant amount of effort.

# Evaluation of the impact of the engineered solution

<ToDo:

* Explain the impact of your engineered solution in a global, economic, environmental and societal context (Use as many as directly apply in your project). For example:
* Evaluate how your solution could contribute to addressing global issues such as climate change, poverty, illiteracy, sustainable development and human well-being.
* Social impact relates to the capacity of your project to create social and cultural change on communities and individuals.
* For environmental, you may evaluate real or perceived health and safety issues associated with the use of your system.
* For economic, you may evaluate the benefits and contribution of your solution to the country’s economic growth and economic competitiveness. Also, you may evaluate the manufacturability of your product (e.g., ease/difficulty of production). Any specific issues and additional costs?
* Use the following table to quantify the significance of your design in the above contexts. You may use a scale of: *low*, *medium* and *high*, where you identify the current status of the contexts as *no impact* and hence show how high your work can impact on them (High being extremely impactful while low being minor impact).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Evaluation Context | Specific Contribution through the Project | level of impact (High, Medium, low |
| 1 | Global |  |  |
| 2 | Economical |  |  |
| 3 | Environmental |  |  |
| 4 | Societal |  |  |

/>

# Conclusion

< ToDo:

* Discuss the main conclusions (e.g., match the project objectives with the achievements in your work and state the degree of achievement).
* Highlight the strengths of the solution and list down its shortcomings (what worked? what didn’t work?).
* Highlight the key contributions and the novel aspects of your work.

/>

# Future work

< ToDo:

Suggested improvements and further work: identify areas of improvement in the project and features of interest that can be added later on. How the solution shortcomings could be addressed? What things could be done better? What additional resources are required to implement the extended / not-implemented design or features?

/>

# Student reflections

< ToDo:

Add individual student **reflections** (**add a sub-section for each student**):

* Lessons learned from the project.
* Professional development you have achieved during your project experience (i.e., new skills gained) and explain its value for your future career.

You can discuss (1) new technical skills acquired such as solving problems, designing and realizing solutions (2) interpersonal skills such as team leading and effective communication (3) personal growth such as adapting to change and acting professionally and ethically.

* Key shortcomings that you should avoid in future projects.
* Key lessons and new attitudes to carry forwards to your professional life from the personal experiences and the team work experienced during the project.

/>

# References

< ToDo: Include the list of references relevant to your work using IEEE citation style. Use research papers and good quality references (such as journal and conference papers from ACM, IEEE and Springer). Do not just use web links. Embed citations of the references in the text of your document.

For further details refer to http://libraries.dal.ca/content/dam/dalhousie/pdf/library/Style\_Guides/IEEE\_Citation\_Style\_Guide.pdf

You can get the PDF documents of good quality papers related to your work from IEEE, ACM and Springer via QU Journal Databases @ http://library.qu.edu.qa/index.php/en/#tab-2

A better way is to search via [http://scholar.google.com](http://scholar.google.com/) (if you do it in QU Campus you can easily get the paper PDF for free).

Remember, always seek your supervisor’s advice and feedback.

/>

# Appendix A – Project Plan

## A.1. Project milestones

< ToDo: List and explain the major milestones of the plan used to deliver the project. Milestones are checkpoints which have specific deliverables to be produced.

It is highly encouraged to adopt an incremental and iterative approach to deliver your project. Each iteration/milestone involves choosing a use case (or a component from your high-level design) and quickly designing, implementing and testing its realization. This way, you can get an early prototype that grows incrementally and converges towards the desired system. You can demo your early prototype to your supervisor to get valuable feedback that can be considered for subsequent iterations.

For the most part, this is taken directly from your interim report, but you should update the plan to reflect what actually happened./>

## A.2. Project timeline

< ToDo: Project timeline defines who will do what and when. You need to:

* Breakdown each project milestone into manageable tasks. Using a table, provide a brief description of each task.
* Estimate the time required to complete each task.
* Allocate the tasks to individual team members and define a work schedule stating the estimated beginning and completion dates of each task. You can use Microsoft Project to create your project timeline. Insert a Gantt chart and/or a project timeline table in this section.

Keep in mind the total number of days budgeted for the project. Note that the role and the responsibilities of each team member should be clearly decided.

Be sure to include time to evaluate your work, to reflect on the experience and to document your experience in intermediate reports at each milestone which will accumulate to form your final report. You should include time to write and review the final report and the final presentation. />

## A.3. Anticipated risks

< ToDo: Present a table of risks highlighting the potential events that might result in failure of successfully completing the project. What is your approach to minimize each risk? />

# Other Appendices

< ToDo:

The following are possible optional appendices you may add to the project report:

* 1. Acceptance testing plan
  2. Any questionnaires, interview questions, etc. used in your project.
  3. Any other appendices may be included to provide supporting details that could aid in the understanding this report.

/>

|  |  |
| --- | --- |
| Check Out 04 (Taken on 21-4-2019, at 8:34 PM) | |
| A close up of a computer  Description automatically generated | A picture containing person, indoor, electronics  Description automatically generatedA picture containing tableware  Description automatically generated |