|  |
| --- |
| CivicTech |
| CivicTech Software Manangement |
| Pricing Documentation |
|  |
| **Smith, Mark** |
| **10/1/2020** |

|  |
| --- |
| This document provides information on what software will be used during development. |

****Contents****

[1. EXECUTIVE SUMMARY 2](#_Toc52899165)

[2. STAKEHOLDERS 2](#_Toc52899166)

[3. REQUIREMENTS 3](#_Toc52899167)

[4. RECOMMENDATIONS 3](#_Toc52899168)

[4.1 Database 3](#_Toc52899169)

[4.2 Data/Relation Modeling 3](#_Toc52899170)

[4.3 Use CASE 3](#_Toc52899171)

[4.3.1 BPMN 3](#_Toc52899172)

[4.4 DEVELOPMENT ENVIRONMENT (WIN) 4](#_Toc52899173)

[4.5 VISUALIZATION 4](#_Toc52899174)

[5. SETUP 4](#_Toc52899175)

[5.1 MEALPLANNER 4](#_Toc52899176)

[5.1.1 Linux 4](#_Toc52899177)

# EXECUTIVE SUMMARY

CivicTech requires a more standardized method of selecting software and environments for developing products for customers. By doing so, it will help alleviate a number of concerns, such as potential software compatibility issues and unintended deprecation, as well as help new recruits jump into projects quicker.

# STAKEHOLDERS

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Role** | **Requirement** | **Date** |
| Sandi | Owner/Administrator |  |  |
|  |  |  |  |
| Mark Smith | Software Developer |  |  |

# REQUIREMENTS

* Software must be able to run on Windows & Linux machines.
* Software must be free OR open source.
* Software must be capable of storing work in a file that can be shared and opened by other users, regardless of operating system.
* Software must be capable of exporting diagram as image or PDF.
* Software is allowed to be recommended if there exists a Community/Free edition, even if it requires creating an account to download.
* Software must not be deprecated.

# RECOMMENDATIONS

## Database

1. PostgreSQL

## Data/Relation Modeling

1. Oracle Data Modeler [Free, supports SQL Server & DB2, Does NOT support PostgreSQL]
2. pgModeler [Free but requires compiling] (https://pgmodeler.io/)

## Use CASE

### BPMN

(https://www.signavio.com/bpmn-introductory-guide/) (<https://www.youtube.com/watch?v=BwkNceoybvA>)

1. Aris Express [requires account] (<https://www.ariscommunity.com/aris-express/installation>)
2. LucidChart? (<https://www.youtube.com/watch?v=BwkNceoybvA>)

## DEVELOPMENT ENVIRONMENT (WIN)

1. Visual Studio Code [Free]
2. MSYS2

SSH Key Generator:

1. https://git-scm.com/book/en/v2/Git-on-the-Server-Generating-Your-SSH-Public-Key

https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent

Required:

1. GitHub

2. Docker

Useful:

1. Introduction To Linux Course [Free] (<https://training.linuxfoundation.org/training/introduction-to-linux-2/?utm_source=lftraining&utm_medium=pr&utm_campaign=linux>)

## VISUALIZATION

# SETUP

## MEALPLANNER

### Linux

1. install virtual box
2. download ubuntu
3. create virtual machine (https://www.wikihow.com/Install-Ubuntu-on-VirtualBox)
4. run through installation
5. restart vm
6. fix screen size (https://blogs.oracle.com/scoter/oracle-vm-virtualbox-61-fix-vm-screen-resizing-on-oracle-linux-8) & (https://ma.ttias.be/auto-resize-screen-size-windows-10-guest-vms-virtualbox/)
7. Update all apps -> sudo apt-get update
8. Install node -> sudo apt install nodejs
9. navigate to /Desktop/mealplanner/meal-planner-fredericton-ui
10. instal npm -> run sudo apt install npm
11. install git bash (https://git-scm.com/downloads) -> apt-get install git
12. create ssh key (https://git-scm.com/book/en/v2/Git-on-the-Server-Generating-Your-SSH-Public-Key)
13. add ssh key to github account (https://docs.github.com/en/enterprise/2.15/user/articles/adding-a-new-ssh-key-to-your-github-account)
14. install docker (https://docs.docker.com/engine/install/ubuntu/)
15. install direenv (https://github.com/CivicTechFredericton/mealplanner/blob/develop/README.md) -> apt-get install direnv
16. Open ~/.bashrc -> gedit ~/.bashrc
17. Add eval "$(direnv hook bash)" to end of file
18. Clone mealplanner (https://www.toolsqa.com/git/clone-repository-using-ssh/) -> git clone git@github.com:CivicTechFredericton/mealplanner.git
19. type "env" to make sure the environment variable "DBNAME" = mealplanner
20. run sudo apt install docker-compose
21. Run docker on mealplanner -> sudo docker-compose up -d
22. setup hooks (cp hooks/pre-commit.sh .git/hooks/pre-commit) & (chmod +x .git/hooks/pre-commit)
23. compile and open mealplanner -> run docker-compose run graphql bash
    1. prompt should change to look something like "root@b12e9d8ef166:/app#"
24. run: createdb -h db -U postgres $DBNAME
    1. creates DB
25. run: ./db-reset.sh (runs script to reset db)
26. run psql -h db -d $DBNAME -U postgres -f seed.sql