|  |
| --- |
| *VENI App* |
| **Feasibility Document** |
| **SE 6387 Advanced Software Engineering Project**  **R.Z. Wenkstern**    ***February 5, 2015*** |

|  |
| --- |
| **Group *X*** |
| **Brian MacKay** |
| **Anant Kambli** |
| **Raleigh Murráy** |
| **Shahed Shuman** |
| **Kathryn Whitmire** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Authors** |
| 1.0 | 16-Dec-2014 | Completed initial draft | Group |
| 2.0 | 05-Feb-2015 | Updated BLAH | Group |

Contents

[Revision History 2](#_Toc404808013)

[1. Introduction 1](#_Toc404808014)

[2. Executive Summary 1](#_Toc404808015)

[3. Background 1](#_Toc404808016)

[4. Alternatives 2](#_Toc404808017)

[5. System Description 2](#_Toc404808018)

[6. Cost-Benefit Analysis 2](#_Toc404808019)

[7. Evaluation of Technical Risk 3](#_Toc404808020)

[8. Operational Impact 3](#_Toc404808021)

[9. Legal Ramifications 3](#_Toc404808022)

[10. Schedule Analysis 3](#_Toc404808023)

[11. Other Project-Specific Topics 3](#_Toc404808024)

[Appendix A: Glossary 4](#_Toc404808025)

[Appendix B: References 5](#_Toc404808026)

# 1. Introduction

In the Veterans Administration (VA) Hospital, veterans and their families spend hours waiting in line to check in for their appointments, before heading to the correct office to wait even longer to actually see the doctor. If they make a mistake, then their entire day was just wasted and they will have to repeat the process another time. The *Veni* system is intended to help address the problems in the current set up in order to make veteran lives easier.

There are two scenarios considered in the making of this system. The first case is for when the patient is at home. The user will be able to start the app and give it his/her information in order to find appointment times, directions to the hospital building, or any special directions they need to follow before their appointment. The second case is for when they are at the VA. The user can start the phone app which will be able to determine if the user is at the building, allow the user to check in for their appointment if it occurs within the next two hours, and get the location of the appointment office.

The app computing environment will include a smartphone app on the front end, a cloud server interface, and the VA database system (henceforth referred to as the *Veni* phone app, *Veni* system server, and VistA server respectively). The current prototype will involve iOS for the phone, a linux Virtual Machine (VM) running in the cloud for the *Veni* system server, and a second VM to fake the VistA server (which will be replaced with the VA’s ‘sandbox’ version at a later date).

There are three major project constraints. The first is the schedule end time of April 2015, which is a hard date for when the prototype must be complete. Second, the team working the project is limited to five people. And third is dealing with the VA in order to obtain permission to access their system. Do we have an alternative third…?

# 2. Executive Summary

See appendix C.

# 3. Background

The current check in process at the VA involves one of two things: a Kiosk System or a receptionist. The system we propose will be similar to the kiosk system in that patients can handle the check-in process themselves, but it will be an improvement in that patients will not need to wait in line for the chance to use the system.

The VistA server containing appointment information is unique to each VA facility. *Veni* will need to request the user’s list of hospitals they visit in order to determine which systems have information on that particular patient. The VA also uses the MUMPS language and data storage system, Java wrapper, and Java-based API for talking to remote systems (such as this one).

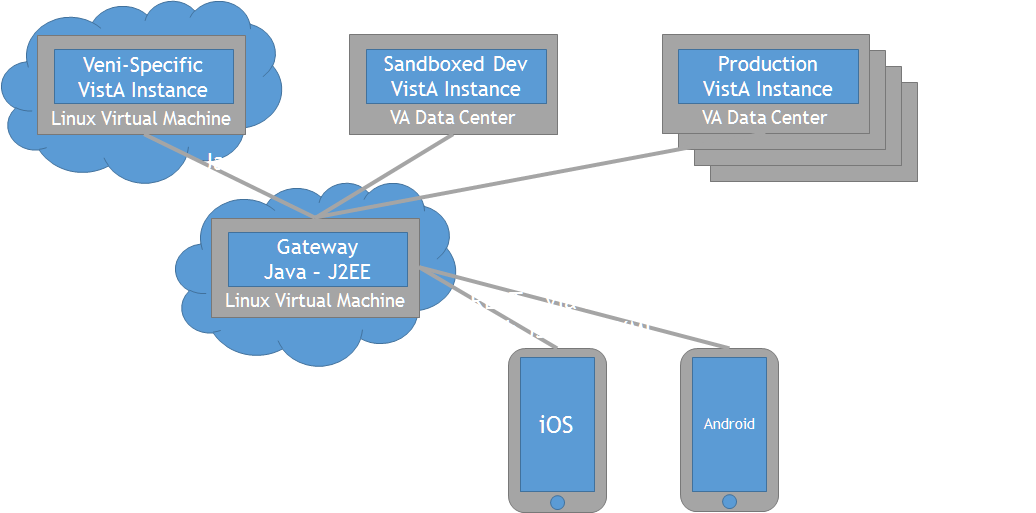
*Veni* will use the phone’s own location, calendar, and mapping services to inform the user of appointment times and give directions. The app will also use geo-fencing to determine the user’s proximity to the VA system. Only when within the geo-fencing will the user be able to check in.

# 4. Alternatives

When planning out the approach for creating the app, there were several choices to be made. In regards to cloud service for creating the *Veni* system server and virtual VistA server, there were the options of AWS or Azure. The phone programming environment could be Native, Cordoba, or Xamarin. All choices were evaluated based on their cost, risk, and compatibility. Free services with which the team had previous experience that could interact well with the other potential components of the system were chosen.

# 5. System Description

The system’s core capability will be the process of checking in for an appointment. Additional capabilities will all branch out from this central concept. The next items to be added will be giving directions to the office of the appointment, getting future appointments added to the calendar, and getting directions to the clinic location.



# 6. Cost-Benefit Analysis

The *Veni* system is a lightweight solution to a difficult problem. Veterans are becoming more tech savvy, so the use of existing technology for this approach will be a welcome change. The greatest benefit will come from the reduction of long lines and longer wait times. The VA will now have another path for the problem besides kiosks or a receptionist, which may allow for the possibility of reducing staff resources needed for patient check-in. This would result in the more effective allocation of staff personnel to other tasks and reduction in the overall costs currently associated with the check-in process.

The cost of this product will be primarily measured in time and effort. The monetary cost will be zero due to the use of free and open source tools. The effort involved for each team member will be at least 20 hours of time per week for the 10 week duration of the project, with a maximum of 30 hours of time per week. Cumulatively, this will be 200 to 300 hours of each individual’s time over the 10 week project (1000 to 1500 hours total).

# 7. Evaluation of Technical Risk

The technical risks of the proposed system involve the security of PHI and PII, communication between the VA system and cloud server through the VA firewall, and determining the best platform for multiple types of user phones.

# 8. Operational Impact

The *Veni* system will be structured in such a way as to have minimal impact on the currently existing VistA database. The *Veni* system server will contain all the functionality of the system, and interact with the VistA server without changing the database content. Essentially, *Veni* will act as an add-on to the currently existing VA operations.

Bullets for here…?

# 9. Legal Ramifications

The security of the system is a very important point. The personal identification information (PII) and personal health information (PHI) must be kept safe from potential hackers. To keep the information safe in the case of a lost phone, the app will require the user to re-log-in every time the app is opened, and the app will close automatically when the phone is locked. The information will also be encrypted when it is sent through the gateway to the VA database or to the phone.

# 10. Schedule Analysis

The current schedule has the deadlines of December 16th, 2014 for the completion of the Executive Summary, Vision, and Feasibility documents and April 3rd, 2015 for the completion of the project. The other milestones which will be interspersed between these dates include, but are not limited to, the software project planning document, the requirements document, the design documentation, implementation and prototype, and the testing document.

# Appendix A: Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| PHI | Personal Health Information |
| PII | Personal Identification Information |
| VA | Veterans Administration |
| VM | Virtual Machine |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Appendix B: References

1. [**www.va.gov**](http://www.va.gov)

# Appendix C: Executive Summary