Document Information

Project name:	Paper Gap Wallet
Date:	28th Jan 2021
Author:	Team 12
Owner:	Jason Quinlan
Document code:	001
Version:	V0.05

Color legend:

Black - older than last week
Blue - updates since last week

Outline Business Case

What is Bitcoin:

Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem. Units of currency called bitcoins are used to store and transmit value among participants in the bitcoin network. Bitcoin users communicate with each other using the bitcoin protocol primarily via the Internet, although other transport networks can also be used. The bitcoin protocol stack, available as open source software, can be run on a wide range of computing devices, including laptops and smartphones, making the technology easily accessible.

Users can transfer bitcoin over the network to do just about anything that can be done with conventional currencies, such as buy and sell goods, send money to people or organizations, or extend credit. Bitcoin technology includes features that are based on encryption and digital signatures to ensure the security of the bitcoin network. Bitcoins can be purchased, sold and exchanged for other currencies at specialized currency exchanges. Bitcoin in a sense is the perfect form of money for the Internet because it is fast, secure, and borderless.

Unlike traditional currencies, bitcoins are entirely virtual. There are no physical coins or even digital coins per se. The coins are implied in transactions which transfer value from sender to recipient. Users of bitcoin own keys which allow them to prove ownership of transactions in the bitcoin network, unlocking the value to spend it and transfer it to a new recipient. Those keys are often stored in a digital wallet on each user's computer. Possession of the key that unlocks a transaction is the only prerequisite to spending bitcoins, putting the control entirely in the hands of each user.

Bitcoin is a fully-distributed, peer-to-peer system. As such there is no "central" server or point of control. Bitcoins are created through a process called "mining", which involves looking for a solution to a difficult problem. Any participant in the bitcoin network (i.e., any device running the full bitcoin protocol stack) may operate as a miner, 1 using their computer's processing power to attempt to find solutions to this problem. Every 10 minutes on average, a new solution is found by someone who then is able to validate the transactions of the past 10 minutes and is rewarded with brand new bitcoins. Essentially, bitcoin mining de-centralizes the currency-issuance and clearing functions of a central bank and replaces the need for any central bank with this global competition.

The bitcoin protocol includes built-in algorithms that regulate the mining function across the network. The difficulty of the problem that miners must solve is adjusted dynamically so that, on average, someone finds a correct answer every 10 minutes regardless of how many miners (and CPUs) are working on the problem at any moment. The protocol also halves the rate at which new bitcoins are created every 4 years, and limits the total number of bitcoins that will be created to a fixed total of 21 million coins. The result is that the number of bitcoins in circulation closely follows an easily predictable curve that reaches 21 million by the year 2140. Due to bitcoin's diminishing rate of issuance, over the long term, the bitcoin currency is deflationary. Furthermore, bitcoin cannot be inflated by "printing" new money above and beyond the expected issuance rate.

Behind the scenes, bitcoin is also the name of the protocol, a network and a distributed computing innovation. The bitcoin currency is really only the first application of this invention. There's a lot more to bitcoin than first meets the eye.

Project Brief
Team 12
Paper Gap Wallet

Project Case:

Our team is aiming to create a new Bitcoin wallet application for creating so-called Paper Cold storage wallets.

As we believe the project will benefit the Bitcoin community greatly by increasing the security of Bitcoin network users and their funds. Especially for new and non-technical users who wish to keep their Bitcoin in cold storage for long periods of time.

The major risk as always with the Cryptocurrency space is the possibility of losing the funds or funds being stolen by a malicious entity (hacker). Our team will try to combat this by keeping up with the latest security standards when developing our application and by keeping the code base as small as possible to reduce the attack surface.

Our app will be first of its kind due to its easy use and security, only one OS with the same purpose as our app exist but it is a lot bigger and much more complicated to use:. https://bitkey.io/

There is also an Mobile application called **Coinomi** which acts as a crypto wallet generator, but unlike our product it does not have the secure air-gapped environment feature that our application holds, and all confidential contents of the crypto wallets are stored on local memory of the device, our product on the other hand will have all contents stored on a removable storage medium (USB) which far exceeds the application in security.

Some Key Terms:

Wallet: Software that holds all your bitcoin addresses and secret keys. Use it to send, receive and store your bitcoin.

Cold Wallet: it cannot be compromised as it is not connected to the Internet. It stores the user's address and private key and works in conjunction with compatible software in the computer.

Paper Wallet: a physical source (e.g barcode on paper, with private/public addresses etc) that acts as the key to the safe that stores your bitcoin

Mnemonic/Bitcoin Seed: Is a 24 word list that stores all of the info needed for a user to recover their wallet. Each wallet has a seed.

Public Address: unique identifier serving as a virtual location where the currency can be sent.

Public Key: corresponds to private key, and compressed into a public address that is shared in blockchain.

Blockchain: records info in a way that is impossible to hack/change as every ledger has all records.

Private Key: allows users to access their cryptocurrency and is proof they own the public key.

Definition

Background:	By solving the Byzantine Generals Problem anonymous cryptographer Satoshi Nakamoto offered an alternative to the modern Central Banking system and economic problems created by the Keynesian monetary policy. Bitcoin blockchain first came online on 3rd January 2009 and it represents the first implementation of fully decentralized peer-to-peer currency protected by Proof Of Work cryptographic technology.
Main Goal:	Build a secure environment for Bitcoin Paper Wallet Generation
Desired Outcomes:	Have a fully functional, easy-to-use app, even for non-technical users
Constraints and Assumptions:	The “Paper Gap Wallet” app will be used by Bitcoin enthusiasts who have a need for a cold storage Paper Wallet. The main constraint is that the app needs to be simple enough that even non-technical users are able to use it securely
Interfaces:	Bitcoin network Other Cryptocurrencies will be added if time permits
Project Approach:	In-house project
Project Product Description:	An App, that creates a live, bootable USB Linux distribution (i.e. Linux distribution that can be booted from removable storage media (USB) instead of installed/booted from Hard Disk Drive) Once booted, the system will present a one-app system that is air-gapped (i.e. network security measure used by computer to ensure that a secure computer network is physically isolated from unsecured networks (public Internet/ unsecured LAN). The User is then welcomed by a start page of the wallet app that takes them through steps to generate wallet seed and public keys.

Project Objectives

	Target	**Tolerance**
Scope	Paper Gap Wallet Application	Some flexibility in final deliverable
Time	22nd March	No late submissions
Cost	n/a	n/a
Quality	Excellent	Nothing but the best
Risks	Security gaps	Community peer review
Benefits	Contributing to the community	n/a

High Level Requirements:

- Python 3.6 Bitcoin libraries
    - our team will use existing well established and tested Bitcoin libraries to create an secure and robust Paper Wallet generator
- Kivy Python Front-End framework
    - our team will use Kivy framework to create simple and clean GUI
- LinuxFromScratch Linux distribution
    - our team will use extended and modified LinuxFromScratch Linux distro to create an air gapped and secure environment

Milestones of Key Deliverables

Milestones	**Finish Date**	**Status**
Idea Brainstorming And Planning	31. Jan 2021	done
Terminal Seed Generator App	07. Feb 2021	done
Front End For Generator App	07. Feb 2021	done
Linux Distro Build	14. Feb 2021	done
Seed Generator App GUI	14. Feb 2021	done
Linux .iso File	21. Feb 2021	done
Documentation	26. Feb 2021	done

Persistent Storage Partition	26. Feb 2021	done
Project Beta Release	07. Mar 2021	wip

External Resources

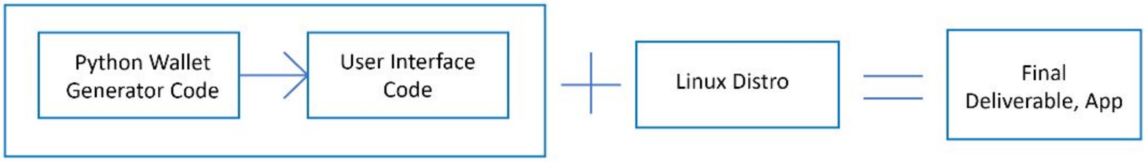
Linux From Scratch: http://linuxfromscratch.org/lfs/view/stable/index.html
Linux From Scratch Live CD Tutorial: https://youtu.be/A9pmmKx0iAU
Tiny Core Linux: http://tinycorelinux.net/welcome.html
Python3 Bitcoin Library: https://github.com/primal100/pybitcointools
Gantt Chart: https://app.agantty.com/#/gantt

Notes (MoSCoW)

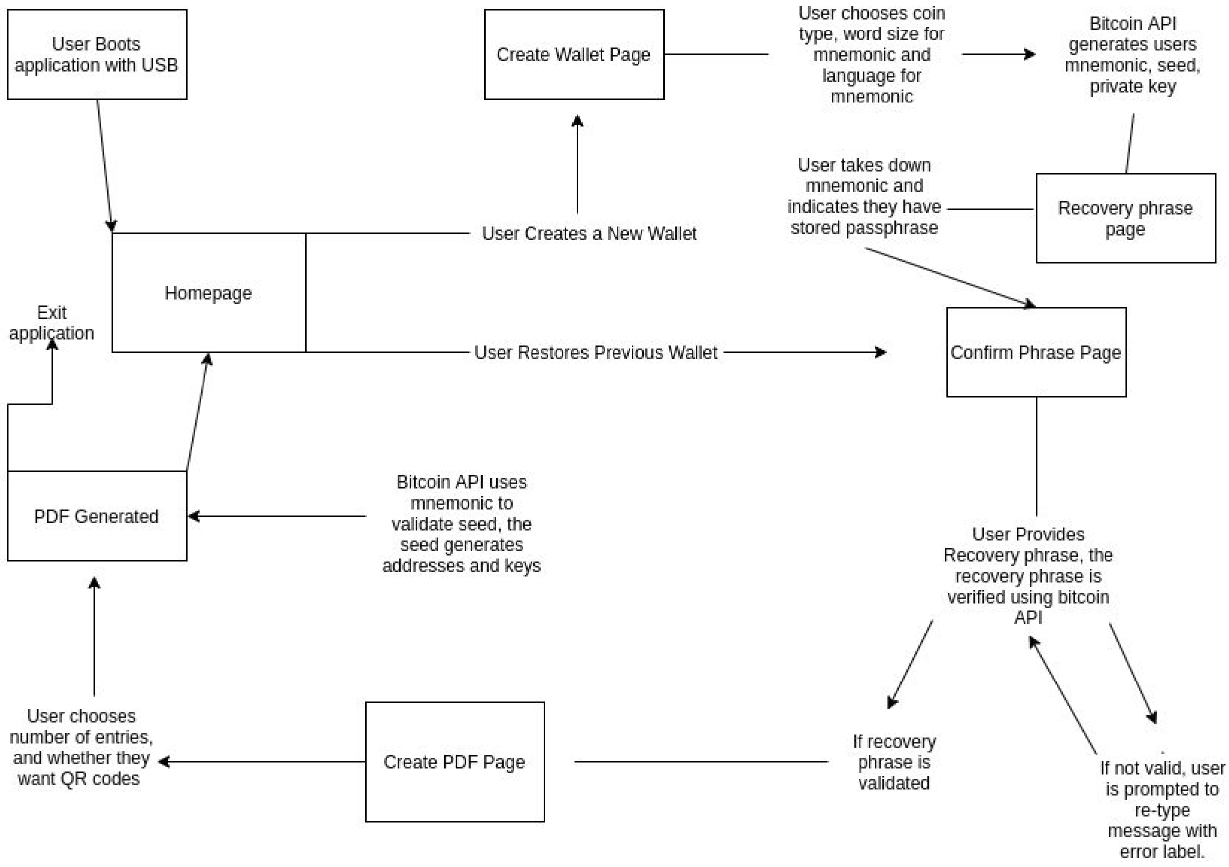
Must Have	- Wallet generation for bitcoin ✓ - A Simplistic User interface ✓ - Creation of wallet ✓ - Restoration of wallet - Air-gapped distro
Should Have	- Responsive UI ✓ - A sized project (within 2gb) - Mnemonic in French Spanish and English ✓ - Availability of QR codes in wallet ✓ - PDF preview of wallet ✓
Could Have	- Creating wallets for ethereum, dogecoin - Ability to create wallets with up to 1000 entries
Won't Have/Would Want	- Wallets for up to 10 different cryptos - Separate application helping user in booting distro containing app

Application Architecture

The entire architecture can be simplified into 3 basic components. The bitcoin library code, the front end generated by the python kivy framework and the air-gapped linux distro. Kivy is used to create a selection of UI elements that make it easy for the user to navigate the difficult process of creating a crypto wallet. Through the actions of each of these UI elements the bitcoin library code gets invoked and generates information to be displayed and calculated throughout the app. External libraries are also used to create things such as QR codes and pdf documents for displaying further information. The linux distro acts as the “vault” for all the private information exchanged within the python application. Once booted up the app is promoted to the user. No information can be exchanged with outside sources. On completion of the wallet creation process all private information gets destroyed. Leaving just the source files for re-running the application (whether to restore or create a new wallet) and any public information created by the app that will be used by the user for crypto transactions (e.g PDF containing wallet info).

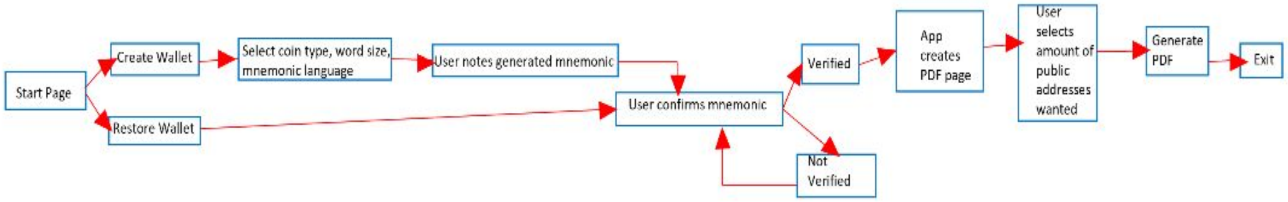


Detailed Architecture:


User Boots application with USB
Create Wallet Page
User chooses coin type, word size for mnemonic and language for mnemonic
Bitcoin API generates users mnemonic, seed, private key
User takes down mnemonic and indicates they have stored passphrase
Recovery phrase page
User Creates a New Wallet
Homepage
Exit application
User Restores Previous Wallet
Confirm Phrase Page
PDF Generated
Bitcoin API uses mnemonic to validate seed, the seed generates addresses and keys
User Provides Recovery phrase, the recovery phrase is verified using bitcoin API
User chooses number of entries, and whether they want QR codes
Create PDF Page
If recovery phrase is validated
If not valid, user is prompted to re-type message with error label.

Application Workflow

Final Deliverable, App Work-Flow


Start Page
Create Wallet
Restore Wallet
Select coin type, word size, mnemonic language
User notes generated mnemonic
User confirms mnemonic
Verified
Not Verified
App creates PDF page
User selects amount of public addresses wanted
Generate PDF
Exit

Project Management Team

Appointee	Role	Description
Ozzak Matic	Master Of The Universe	Linux development and project management
Oliver Cunnington	Technomancer	Linux development and Linux distro integration
Robert O'Brien	Grand Byte-Wizard	FrontEnd development and integration using Kivy python framework.
John Furlong	PyLord	Python development and blockchain operations in charge of back end wallet generation.
Janine Dunlea	Code Empress	Python development and blockchain operations in charge of back end wallet generation.

Detailed Job Description

Ozzak Matic was working on the Linux side of the project. He started by learning about the Linux packages and building the Linux file system and dependencies mostly using the “make” command and making sure it all fits together. The tutorial used for the second week was LFS - Linux From Scratch. Together with Oliver Cunnington he built our own custom Linux distribution without networking capabilities to achieve the goal of having the air-gapped and safe environment for our application. After this subteam had the basic operating system done Oliver continued with GREP file bugs to make sure we can boot into it. And Ozzak went over to try and make it live bootable from the stick. Making a bootable .iso file proved to be more challenging than expected and this functionality is still under development.
On the other side Ozzak was the main person in charge of documentation and project brief. This was the whole team effort but the outline and direction was done by Ozzak.

Oliver Cunnington was responsible of Core Implementation of Linux Kernel that:
Boots from a USB stick with no networking drivers implemented.
Restricted to user mode 2 or 3 (Single-User or Multi-user no networking)
Blacklist drivers for networking (/etc/modprobe.d/blacklist.conf)
Capable of writing to a persistent storage partition of the memory stick
Launching kivy python GUI/Application on launch
Potentially go on to create "boot-stick maker" tool

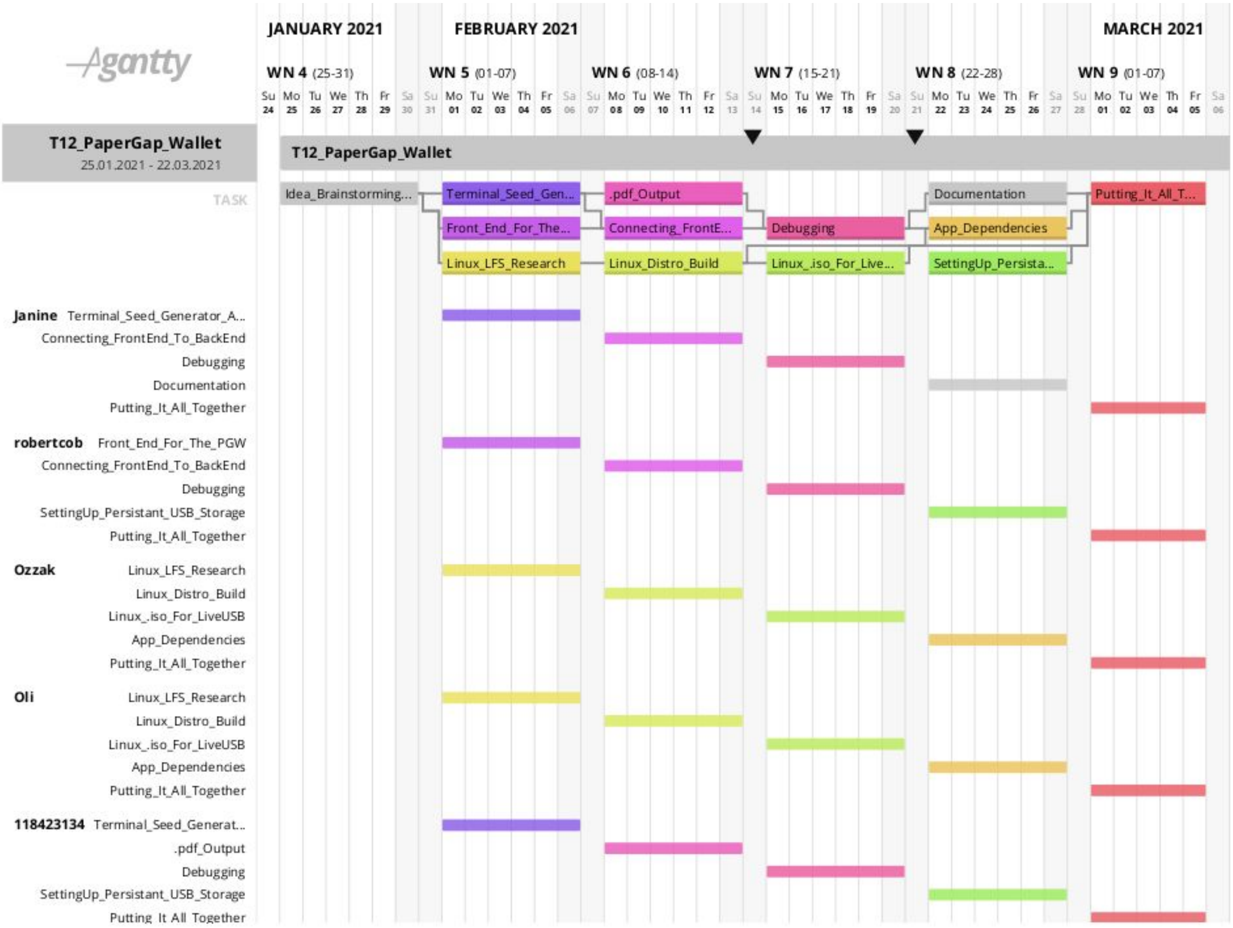
Robert O'Brien was working on #TODO

John Furlong was working on #TODO

Janine Dunlea was working on the python development for generating wallets with mnemonics, keys, and addresses, along with John. The first week consisted of familiarizing myself with what Bitcoin is and it's key concepts, as it was new to me. Videos from Andreas Antonopoulos helped greatly with this, all available on youtube. I then studied and implemented the bitcoin libraries available on python and the best methods for creating mnemonics and keys that would ensure security and randomness. I did this through PyCharm. I am also keeping minutes of any team meetings in order to keep track of progress in our project.

Gantt Chart

Whole Project Overview



<u>Ghand Chart Task Description:</u>

1. **Idea Brainstorming**: whole first week to be used for brainstorming and idea development
2. **Terminal Seed Generator App**: back end for the wallet generator application
3. **Front End For The App**: develop GUI for the wallet generator application
4. **Linux LFS Research**: “Linux From Scratch” research
5. **.pdf output**: back end application generated public keys to be printed in a .pdf file
6. **Connecting FrontEnd And BackEnd**: integrating front and back end of the application
7. **Linux Distro Build**: building our own distribution of Linux
8. **Debugging**: testing and debugging
9. **Linux ISO File**: making our own distribution of Linux live bootable from the USB
10. **Documentation**: documentation and user manuals

11. **App Dependencies**: sorting dependencies between Python, Kivy and our Linux distribution
12. **Setting Up Persistent Partition**: creating persistent partition for the pdf file
13. **Putting It All Together**: integrating all the elements together

Week 6 Overview - Putting it all together

#TODO