

Chillieman: Empowering the 99% with Automation

The purpose of this document is to explain the functionality of ChillieToken, and the associated ChillieWallet application. A general roadmap will also be provided, and a bit of information about my situation, in order to be as transparent as possible.

If you are reading this White Paper, that means the Token's Smart Contract is just about ready for launch! While the Token may be in a good position, and ready to interact with the Chillie Wallet app – There is still much to get done on the accompanying Wallet.

I cannot provide solid delivery dates when things will be complete on the Application side. I am a professional Software Engineer, specializing in Android development, and I'm taking this Project on completely by myself. If the launch of the token on BSC goes well, I may be able to direct all my time towards development of the App. If things don't go so well, I'll be forced to stick with a salary job, and my time will not be as available, causing development of the App to drag.

The End goal for ChillieToken is to Fuel an application that gives the 99% the power of automation! There are plenty of the rich 1% out there with the resources to run sophisticated bots that do their trading for them. I will provide this functionality to the masses, for a very low fee. The wallet user will only be taxed after they have profited off the trade. Chillie Wallet only profits when the user profits! I intend to empower the "average joe" to be able to hold their own against the people of Wall Street – This is the motivation behind ChillieWallet and the fueling token, ChillieToken!

Chillie Token:

ChillieToken is an ERC-20 token that is inspired by DeFi projects that automatically add liquidity by taxing the transfers of the token. Most of the Smart Contract is based on Open-Source code found in <https://github.com/OpenZeppelin/openzeppelin-contracts>. For the token code itself, I started with an existing, very popular, ERC-20 token. This token included functionality for reflections, token burning, and automatically providing liquidity. I removed all code related to Burning Tokens, and Distributing Reflections, and just kept the Automatic Liquidity as my starting point.

***Note that the Token burn was achieved by sending 50% of initial supply to a NULL address, causing half of the Reflections to end up burnt. If you think about that... that means that over 50% of the fees that were supposed to be collected for Reflections (for the community) end up in the trash... *cough* bad programming *cough**

I then created my own logic for Token transfers and implemented a straightforward function for collecting taxes. A ChillieTransfer enforces a maximum token limit for accounts. If an address is attempting to receive tokens, and it would bump their token balance above 0.1% of the total supply, the transaction will fail. The only exception to this rule is reserved for the Smart Contract, Chillieman, and any Exchanges. The Owner will have the ability to add exchanges to the list that is excluded from the max token limit. Any time an Exchange is added to the list, the Token will emit a **ExchangeAdded** event to retain full transparency. **ExchangeRemoved** event will be emitted if an exchange was removed from this special status.

I analyzed the code that was responsible for adding liquidity and found a huge problem. The liquidity tokens were being purchased with the Owner listed as the recipient, giving the Owner instant access to all that liquidity. I analyzed the web and found another huge problem. UniCrypt wants 15,000 USDC to create a dynamic vault for the Liquidity Tokens... Since I'm a programmer, it just wouldn't feel right paying all that money when I can just implement a tiny bit of extra logic to get the Smart Contract to lock its liquidity.

To resolve this Liquidity issue, ChillieToken assigns itself as the receiver of the liquidity tokens, not the Owner. With each new batch of liquidity, a timer is set (or reset) to 1 year until

any liquidity can be claimed. The only way to get the liquidity out is to not have meaningful token activity for a full year – Which I hope never happens, so that Liquidity should never see the light of day! If the liquidity is ever claimed, the Contact will emit a **LiquidityUnlocked** event to retain full transparency. Liquidity is added after the Liquidity stash (collected from transaction taxes) has grown larger than 0.01% of total supply (100 million). Liquidity is also added when the Chillie Wallet Fee stash is processed.

I stripped out any code that gave the Owner the ability to change the Fee Rates, so fees can never be raised. I added new logic that allows Chillieman to turn the fees on and off globally (for everyone) which may come in handy in the future for events, holidays, and possible migrations. I added a lot of functionality that is designed to be used by the Chillie Wallet. A function for paying ChillieWallet Fees, and a function for processing those Fees, further fueling the liquidity of Chillie Coin. The two projects will constantly fuel the success of the other, causing a self-sustainable system, if people are using the Chillie Wallet! I will do everything in my power to make Chillie Wallet irresistible to ensure the success of the Token.

...So, um... why should I trust you?

If it's not clear by now, I want to make this explicit: **This token and wallet are designed to be as fair as I can possibly make it.** Trust is one of the hardest things to gain, especially in the crypto space, as the number of scams out there is just ridiculous. Even though I have gone to great lengths to make the Smart Contract code as fair and safe as possible, I understand that it may not be easy, for those who aren't a geek like I am, to see that the Contract is fair and safe.

I will fight to gain your trust; any reasonable way I can. I am making it a priority to get a contact Audit as soon as I can, but I assume that's going to be a bit expensive and may have to wait until some Token Taxes (Or Donations) are generated so I can pay for it. I will keep up communication with the community, constantly providing Dev Updates and Sneak Peeks. I will embrace feedback from everyone, even my biggest critics, and address any issues that arise. Slowly but surely, you will see that I can be trusted. Transparency is in my nature, and I am genuinely interested in helping other people make money – **So let's do this!**

Token-omics:

1. Binance Testnet
 - a. Name: **ChillieTestToken**
 - b. Symbol: **cTST**
 - c. Decimals: **9**
 - d. Total Supply of Token: **1 billion**
 - e. Initial Supply to PancakeSwap Clone: **100% + 0.001 tBNB**
 - f. Router Address: **0xD99D1c33F9fC3444f8101754aBC46c52416550D1**
 - g. Contract Address: [**Chillie** – To Update]
2. Binance Smart Chain
 - a. Coming VERY soon!
3. Fees: **10%**
 - a. 9% will be taken to automatically provide liquidity
 - i. Liquidity will automatically be locked in the Chillie Token contract for 1 Year. The 1 Year timer is reset every time the Liquidity stash is converted into LP Tokens.
 - b. 1% will be taken to provide for Chillieman
 - i. This tax will Fuel Chillieman, which will fuel ChillieWallet, which will fuel ChillieToken.
 - ii. I think I deserve something for my hard work =]
 - iii. This will be turned off once the Chillie Wallet launches!
4. Maximum Token Amount: **0.1% of Total Supply**
 - a. I have designed ChillieToken in such a way that standard accounts cannot hold more than 0.1% of the total supply of tokens.
 - b. While a standard account cannot purchase over 0.1%, they still can receive Token Rewards by using the ChillieWallet. There is no limit to the amount of Token Rewards you can earn by using the ChillieWallet!

- c. Exchanges are an exception, and must be able to hold more supply, as they typically hold the Tokens for their customers, until the customer decides to Withdrawal their tokens to a personal address.
 - d. Exchange Exemptions can be removed or added by Chillieman as needed, which will allow the address to hold more than 0.1% of supply. The first Exchange will always be the UniSwap V2 clone related to blockchain which the ChillieToken was deployed. (i.e., PancakeSwap)
5. Burn: **None**
- a. The only tokens to be burned will come from my personal development address, and will manually be sent to 0x00000000000000000000000000000000fAdED
 - b. I strongly encourage you to NEVER burn your personal tokens for any reason
6. Whale Protection:
- a. Individual Whales will have a tough time controlling this Token!
 - b. First line of defense is the maximum token amount that a standard address (not an exchange) can hold. While this doesn't make it impossible to control, a Whale would need 100 different addresses to hold 10% of the total supply, which seems like more trouble than its worth.
 - c. There will be no Reflections, as they overly reward the largest holders of the token. The rich get richer, while the masses only receive extra pennies.
 - d. As the token is constantly funneling new Liquidity into the original exchange, it becomes increasingly harder to control the Token Price by adding and removing personal Liquidity.
7. Ownership (**onlyChillie**):
- a. I have absolutely no intension of revoking ownership to this Contract. Its hard coded into the contract that I am the only "owner", and there can never be a new one.
 - b. I took out most benefits of ownership, so there's no immediate danger for me being the owner:
 - i. Liquidity is not assigned to the Owner like other tokens

- ii. It is impossible to Mint new tokens.
 - iii. Tax Rates cannot be changed in a harmful manor
 - c. I'm going to explain the functions that can only be called by Me, so hopefully it's not as scary with me remaining the owner, I think I'm a pretty fair one!
8. Functions that can only be called by Chillieman:
- a. **suspendTaxes** - Turn off All Taxes for Everyone
 - b. **resumeTaxes** - Turn on All Taxes for Everyone
 - c. **addExchange** - Add an Exchange Address (Can hold more than 0.1%)
 - d. **removeExchange** - Remove an Exchange Address (Cant hold more than 0.1%)
 - e. **removeChilliemanTaxExemption**
 - i. When ChillieToken is created, Chillieman will be exempted from Taxes by default (so no tokens are lost when adding the Initial Supply)
 - ii. Once all Tokens have been supplied, this function will be called, and then Chillieman will be taxed on all future transactions, just like everyone else.
 - iii. Emits a **ChilliemanIsNowTaxable** event to prove this function was called.
 - f. **claimDevelopmentTax** - Claim the Dev Stash for all my hard work
 - g. **discontinueDevTaxes**
 - i. Stop collecting 1% tax for the Dev Stash and set the full tax to be sent to the Liquidity Stash.
 - ii. **Yes, I intend to stop collecting Dev Taxes on the Token once the Wallet is released**
 - h. **startChillieWalletSeason**
 - i. Start a new Chillie Wallet Season!
 - ii. Set the ChillieToken price that the Chillie Wallet must pay to use Chillie Wallet.
 - i. **endChillieWalletSeason**
 - i. End the current Chillie Wallet Season!
 - ii. Clears the list of users who are authorized to use Chillie Wallet.

What are the features of Chillie Wallet?

Privacy is number one, ChillieWallet will never capture any personal information about you such as Email, Real Name, etc.

The idea of Chillie Wallet is to turn your phone into your own, full-time, day trader. The focus area is Tokens that are listed on a Decentralized Exchange such as UniSwap or PancakeSwap. One of the most annoying things about these exchanges, is there is no meaningful method of creating Limit Orders! It also can be annoying using the website interface of the DEX, especially when using a phone. With the ChillieWallet, you can be faster than the competition because the App is communicating directly with the Block Chain and Smart Contracts and will use the preprogrammed logic that you teach it to use.

Wouldn't it be nice if you could tell your phone to Buy a certain Token once the price dips to your target? Wouldn't it be nice to have a Stop Loss on your tokens, that way your exit strategy can automatically be executed when the market is crashing? Wouldn't it be nice to give your phone precise instructions on how it should Flip a certain token, and it just does it for you? And wouldn't it be nice to use this functionality on any DEX on a blockchain, even ones that you can manually add yourself?

Another big plan of implementation is to have robust vetting of any token that is purchased using ChillieWallet. The ChillieWallet v2 will be focused on detecting common scams, and clearly warn the user before they buy something that they will most likely not be able to sell back. I have been scammed, rug pulled, honey potted, and lied to. I want to provide a tool that can mitigate this damage.

Much more information to come on the App! Keep your eyes on <https://chillieman.com> for continuous updates on the ChillieWallet. I have not figured out what will be my main method of communication as it relates to Development of the Chillie Wallet, but I will make it obvious on the site so you can stay up to date!

Project Planner

- ~~1. Create ChillieToken~~
 - ~~a. Automatically Generate Liquidity~~
 - ~~b. Collect taxes in a straightforward way.~~
 - ~~c. Buy / Sell Tokens from Router (v2 Routers)~~
 - ~~d. Recycling — DO NOT let a shred of ETH or Token go unaccounted for~~
 - ~~i. Seeing a lot of waste in other Token Contracts~~
 - ~~ii. Example: Check for left over eth or tokens after adding liquidity — Use the leftovers next time you are trying to add liquidity!~~
 - ~~e. Make it obvious that it's a Chillie Contract~~
 - ~~i. If you read the code, you will know what I mean, lol~~
 - ~~f. Add Functions allow integration with the Chillie Wallet~~
 - ~~i. Start / End Seasons~~
 - ~~ii. Charge / Process Chillie Wallet taxes~~
 - ~~iii. Put in logic to prevent people from using Chillie Wallet for Free~~
2. Test on BNB Testnet (Ongoing as of 5/30/2022)
- ~~3. Create White paper~~
4. Create an initial version of the ChillieWebsite
5. Create social media presence
 - a. Discord
 - b. Twitter
 - c. Telegram
 - d. Facebook
 - e. Reddit
 - ~~f. Github~~
6. Launch Token on Binance Smart Chain
7. Get Token listed on CoinMarketCap and CoinGecko
8. Release Chillie Wallet Alpha / Beta
 - a. Connect to BNB Testnet
 - b. Create Pincode & Passphrase (Authentication)
 - c. Create wallet
 - d. Use of one wallet
 - e. Fetch Balance, Transfer tETH
 - f. Import Tokens, Fetch Balances, Transfer Tokens
 - g. Generate QR codes for easy addresses sharing.
 - h. Track Token Prices – Get Notified when a certain price is hit (high or low)
9. Improve the Website

10. Release ChillieWallet v1
 - a. Improved Security (AES-256 / SHA-256)
 - i. Database file encryption
 - ii. Database entries encrypted
 - iii. Wallet Credentials encrypted using Device Specific Metadata
 - b. Import wallet (From Private Key or from Seed Phrase)
 - c. Create/Import many Wallets
 - d. Connect to multiple blockchains (Smart Chain & Ropsten)
 - e. Utilize PancakeSwap & ChillieToken
 - f. Set up Tasks / Workers:
 - i. Buy a token when it dips below a target price
 - ii. Sell a token once it dips below a target price (Stop Loss)
 - iii. Sell a token once it raises above a target price (Take Gain)
 - g. Generate Taxes to be sent to ChillieToken
11. Revisit the Token Contract, enhance so it can be used with v3 Liquidity Pools
12. Launch Token on ETH Ropsten Testnet & Release ChillieWallet v1.5
 - a. Connect to ETH Testnet
 - b. Integrate with UniSwap (v3 Routers)
13. Launch token on Ethereum Mainnet & Release ChillieWallet v2
 - a. Connect to ETH Mainnet
 - b. Allow user to choose which Swap to use (Import its address and use it!)
 - c. Improved Automation: Fully Customizable, easy to create scripts, where you can teach your phone exactly how to flip a token – Users will fully achieve Automation, they just need to provide a solid trading strategy
14. Launch token on other EVMs (i.e., MATIC, AVAX, KuCoin)
 - a. Update ChillieWallet to connect to any EVM Blockchain the user wants.
15. Expand team and start looking for talented people to bring on board
 - a. Web Engineer to create an exceptional website
 - b. iOS Engineer to team up with to create an apple version of ChillieWallet
 - c. Marketing team (I'm good at Coding, not good at this lol)

Version History:

[v1] - Chillieman - 5/30/2022: Create Initial Version of this Document

TODO:

- Create and add Flow Diagram for transactions (Illustrate different Stashes, Taxes, Fees between Chillie Token and Chillie Wallet).
- Update Contract Address for Testnet and add Mainnet information.
- Reach out to the community and see what they think about this White Paper.