

# Mobile Engineer Interview Project (Take Home Test): Penguin Pay

## Brief: A simple app to help launch PenguinPay in Binaria

You work for an imaginary money transmitter, PenguinPay. Penguin pay sends money from the US (the sending market) to several countries in Africa (the receiving markets), where it pays out to different delivery networks such as M-Pesa in Kenya and MTN in Uganda. As part of PenguinPay's growth strategy, they are looking to launch a new sending market, the (fictitious) country of Binaria.

Binaria is home to multiple diasporas who are eager to send money to their friends and families back in their home countries. Binaria has one main requirement for remitting money, one which PenguinPay will need to take into account in their app: residents of Binaria are only allowed to use binary numbers.

For example, where in the US a user would enter "22" if they wished to send \$22, in Binaria they must enter "010110". If a resident of Binaria wishes to see how much money in local currency the recipient will receive, it too must be displayed in binary. For example, entering '010110' for a recipient in Nigeria should show '01111100010001' (\$22 converted to NGN at the current exchange rate of 1 USD = 361.50 NGN). Binaria has pegged its currency to the US Dollar.

Your job is to design a prototype of the 'Send Transactions' screen for PenguinPay's upcoming launch. The PenguinPay Product Manager has given you the following requirement:

- Allow the user to enter the first and last name of a recipient.
- Allow the user to select the recipient's country from a list of Kenya, Nigeria, Tanzania, and Uganda.
- Allow the user to enter the phone number of the recipient. It should be validated based on the recipient's country.
- Allow the user to enter the amount they wish to send in binary.
- Display to the user the amount the recipient will receive in binary with the appropriate local currency indicated. Don't worry about handling fractional binary amounts.
- Allow the user to press send, at which point an appropriate message should be given to the user that their transaction is being sent.
- The amount the recipient will receive should be converted from Binarian Dollars to local currency at current market rates. As Binarian Dollars are pegged to US Dollars, you can use US Dollar (USD) exchange rates.
- Current exchange rates can be accessed via APIs from [openexchangerates.org](https://openexchangerates.org) (free plan) using the latest.json endpoint (the other endpoints are not available on the free plan).

## About the Task, Submission, and Timelines

If you have any questions about the requirements please don't hesitate to ask.

Please do not spend longer than 3-4 hours on the task. Many applicants find that is not enough time to do everything. Don't worry, this is intentional! Please prioritize as you see fit and include a note explaining any time-saving compromises you made.

**We will try our best to review the app and your code within 5 working days. Thereafter we will schedule your final round of interviews.**

### Please Note:

1. **For iOS candidates: Please build the app in Swift**

## 2. For Android candidates: Please build the app in Kotlin

### Evaluation Criteria

When we evaluate your app, we'll mainly consider the following criteria (in order of importance):

- Is it bug-free?
- Is the code easy to understand, maintain and unit test?
- Is the UI/UX usable?
- Does it meet all of the requirements?

The UI doesn't need to be fancy, but it should exhibit some level of thoughtfulness and polish such that it is intuitive and just works. If the user does something which does not make sense, and there's always at least one such user, the app should not crash.

It's up to you how you would like to send us your app, but we'd like to be able to run it and we'd like to see the code behind it.

### Useful information

	A	B	C	D
1	Country	Currency Abbreviation	Phone prefix	Number of digits after prefix
2	Kenya	KES	+254	9
3	Nigeria	NGN	+234	7
4	Tanzania	TZS	+255	9
5	Uganda	UGX	+256	7
6				

---

**As a company, we value:**

- **Prioritize fearlessly**
  - *Sendwave team members recognize there will always be more problems than team members to solve. We are comfortable letting the little ones continue to exist while obsessing over the few that matter most.*
- **Take full ownership of the outcomes that you're responsible for**
  - *A Sendwave team member's job is not to do what they can to solve a problem. It's to ensure that problem is solved. If an attempt to solve a problem doesn't work, we seek an alternative. If we need help, we request it, and, if necessary, demand it.*
- **Figure out your root problem, and work towards a solution that addresses it**
  - *Sendwave team members can only prioritize effectively and take real ownership if they deeply understand the underlying cause of the problems they're trying to solve.*
- **Test your ideas early & often**
  - *Sendwave team members view roadmaps as ever-evolving drafts and welcome opportunities to discover that what they had in mind won't work so they can move to an even better roadmap.*
- **Embrace embarrassing honesty -- from yourself and others**
  - *The Sendwave team functions best when we're open and honest with one another - especially about our challenges and doubts. We lean into uncomfortable conversations and support our colleagues*

*when they do the same.*

- *Maximize for energy, not time spent*
  - *Sendwave team members measure our own and others' contributions to the company by work completed, not time spent working. We believe work/life balance is compatible with, even necessary for, high achievement over the long term.*