The Mifos Initiative: GSoD’2022 Case Study

Update Platform and Collaboration Documentation for Launch of Mifos X 3.0

Organization or Project:

The Mifos Initiative
(https://mifosforge.jira.com/wiki/spaces/RES/pages/3038248961/Season+of+Docs+-+2022)

Organization Description:

The Mifos Initiative is a global 501(c)3 fintech non-profit leveraging the cloud, mobile & open source community to democratize financial services worldwide and digitally transform the world’s 3 billion poor and underbanked. Mifos has pioneered open-source banking technology for the past fifteen years transforming the entire sector at each major stage of evolution from microfinance to financial inclusion to digital financial services and now embedded finance. Mifos guides the open-source community, steers the roadmap, and stewards the vibrant ecosystem of organizations building solutions on its open platform. Their building blocks for banking, recognized as digital public goods, make core banking commoditized infrastructure, empowering any organization, anywhere to embed any financial service to any customer via any channel. These building blocks provide the common functionalities for creating customers, managing wallets, savings, and loan accounts, orchestrating payments, and maintaining the financial ledger & reports. More than 20 million clients are reached by 500+ financial institutions across 41 countries using solutions powered by its APIs.

Authors:
Edward Cable & Tapasweni Pathak

Problem Statement

This project will build off of the 2021 Season of Docs project which consolidated our documentation into one unified portal with separate spaces for developer and user documentation.

With this solid documentation foundation in place, we now need to update our documentation to reflect the major transformation of our project at both an architectural as well as community infrastructure level. In 2022 we are releasing the most substantial evolution to the platform since its initial launch nearly 10 years ago. This major release will be titled Mifos X 3.0 built on top of Fineract 2.0.

Throughout 2021 and 2022, the community has placed a renewed focus on the Fineract 1.x codebase with major architectural refactorings to enable greater scalability and modularity. In addition we have released a redesigned and modernized version of the reference user interface for staff on Angular, our web app. Lastly our payment orchestration engine, Payment Hub EE has achieved production readiness and is now a more integral component of the end to end stack. . In addition to these enhancements at the platform, UI, and orchestration layer, we have significantly improved deployability through containerization, the contribution process through more automated code checks and analysis as well as the QA process through implementation of the Cucumber test framework.

This year's project will focus on updating and drafting documentation to reflect this next major evolution in our platform and community infrastructure.

- The new refactored architecture of the Fineract 1.x platform itself including changes such as improved read/write separation of API calls, migration of database from MySQL to PostgreSQL, migration of ORM from OpenJPA to Eclipselink, introduction of Kafka for reliable event handling framework, transactional idempotency, and integraiton of Spring Batch for improved end of day processing.
- The updated reference web app user interface with modern components, material design, and skinnability built on top of Angula.
- The production-ready payment hub EE with additional payment connectors, enhanced operational interface, dashboards and visualizations, and notifications module.
- Greater deployability through improved containerization, terraform scripts and configurable Helm Charts
- Streamlined contribution process with more automated code checks
- Improved QA infrastructure through Cucumber Test Framework.
- Documentation of Mobile UI library and SDK Components (Optional scope)

The more unified we can make our documentation and show how the various components and solutions all fit together via live APIs and sandbox environments, the more innovation and upstream contribution we can catalyze.

Proposal Abstract

The project will consist of refining existing documentation written by developers and writing new documentation. A small portion of the project will focus on ongoing optimization, organization and consolidation of documentation.
As part of the development process, our developers include a baseline level of technical and functional documentation. The technical writer chosen, will work alongside existing community volunteers to make the existing documentation more robust and draft new documentation where it doesn't exist.

Project Description

Creating the proposal

Documentation audiences across our community include the end users of our software, the staff of financial institutions and fintechs that use Mifos to create customers and their accounts, configure loan and savings products, process transactions and track repayments and deposits, manage their general ledger and generate financial and operational reports. The primary audience of our documentation is our partner community who host, deploy, configure, and support the software for these financial institutions, build new fintech solutions and applications using the APIs, and develop, maintain, and extend the core upstream project which powers their solutions. Joining these partners as part of our developer community are individual volunteers, interns, and corporate strategic partners who collectively guide the development, QA, and release management to ensure timely and high-quality open source releases of our platform.

Budget

Total project budget is $15,000 USD

Item	Budgeted (USD)	Actual (USD)	Notes
Technical Writer	$12,000	$12,000	
Graphic design - architecture diagrams, flow diagrams and visuals to accompany documentation	$750	$750	
Project Management, Supervision, and Review of Deliverables	$1,500	$1,650	To cover partial cost of scoping out project, managing its delivery and final review of deliverables.
Volunteer Stipends (2 at $300 each)	$600	$600	For volunteers that will be closely providing information and/or reviewing deliverables
T-Shirts for Volunteers	$150	0	Printing of t-shirts for documentation volunteers

Participants

- **Technical Writer -**
- **Project Management**
  - **Edward Cable**
  - **Mihaly Dallos**
  -
- **Subject Matter Experts**
  - **Aleksander Vidakovic**
  - **Victor Romero**
  - **Bharath Gowda**
  - **Arnold Galovics**

Proposed Timeline

The project consists of refining existing documentation written by developers and writing new documentation. A small portion of the project will focus on ongoing optimization, organization and consolidation of documentation.

As part of the development process, our developers include a baseline level of technical and functional documentation. The technical writer chosen, will work alongside existing community volunteers to make the existing documentation more robust and draft new documentation where it doesn't exist.

Architectural Documentation (10 person days)

Major refactorings have occurred at every layer of the platform architecture - the documentation must reflect the changes and the impact it has upon development, deployment and performance.

- Refine documentation on the enhanced ability to separate read/write API operations for greater performance (1 day)
- Refine documentation for migration of database from MySQL to PostgreSQL including the migration from Flyway to Liquibase for database migration scripts (1.5 days)
- Refine documentation for tuning and indexing to optimize performance based on PostgreSQL and Eclipselink enhancements. (1 day)
- Write new documentation on transaction idempotency and business and effective date configuration (1 day)
- Refine documentation on new reliable event handling framework built on Kafka (2.5 days)
- Refine documentation on integration of Spring Batch for parallelization of batch jobs, running smaller chunks of batch jobs, ensuring no transaction conflict between real-time and batch processes including architecture of new framework, how to configure and schedule jobs, etc. (3 days)

Functional & Deployment Documentation (12 person days)

- Draft new documentation outlining functionality of payment hub, each of its new modules, and use cases it supports (3 days)
- Draft new documentation on how to install, deploy and configure Payment Hub EE (2 days)
- Update documentation and screenshots in user manual to reflect design of new web app (2 days)

- Update and refine documentation to reflect new DevOps tooling for deployment of the platform and Payment Hub EE including Terraform scripts, Configurable Helm Charts, Kubernetes and Containerization Optimations (3 days)
- Update critical documentation that is outdated or not clear including installation guide, reporting, SMS gateway configuration, and self-service user creation. (2 days)

Collaboration Documentation (7 person days)

- Draft documentation related to automated code checks, static analysis, and other build-level automation that has been introduced to improve code quality and reduce friction of upstream contribution (1 day)
- Update documentation on upstream contribution process and ensuring that contributors know how to properly configure Github to sync their downstream codebase with upstream project and do development in an upstream-first manner that is aligned with Apache Way (2 man-days)
- Draft documentation for QA/testers on how to write new tests and leverage the Cucumber Test framework (2 man-days)
- Draft documentation for developers on how to write code leveraging new Cucumber test framework for unit testing (1 man-day)
- Update FAQs explaining difference between Mifos and Fineract. (1 day)

Standardize Documentation (5 days)

For each of our projects we have a wealth of documentation that needs to be organized and arranged according to a standard format/flow:

- Define standard for existing documentation and ensure each project's user and technical documentation adheres to that standard format/flow:
  - For end-users - installation, troubleshooting common issues, system specs, tutorials, product overview, quick start guide, domain/topic explainers. (2.5 days)
  - For developers - contributing guidelines, project setup guide, troubleshooting tips, architectural description/overview, release notes/history. (2.5 days)

Review & Incorporate Feedback & Draft Case Study (3-5 days)

- Based on review of the above deliverables by project supervisor and volunteers, technical writer will implement feedback and refine deliverables
- Draft season of docs case study

Project Total: 50 - 55 person-days

Summary

Our Season of Docs 2022 experience was very productive in terms of the output and content that our technical writer, Tapasweni Pathak, was able to complete. We now have a much more solid foundation from which to build upon but still need to mobilize our community to consolidate and streamline documentation so specific personas and stakeholders can achieve what they're looking to do in a more efficient fashion. Independently we made great progress on each of the intended deliverables and now need to unify all these efforts so they can be easily accessed in a cohesive fashion. Apart from the documentation that was created, we were able to improve upon the documentation creation process by incorporating into our development and version control flows on Github.

Some of the major milestones that were achieved:

- **Architectural Documentation**: Tapasweni was able to work in parallel with the ongoing architectural and feature development efforts to deliver more polished and comprehensive documentation that will help our community be aware of the major architectural changes forthcoming in our new versions.
- **Collaboration Documentation:** While we still have progress to be made around Getting Started documentation for developers and implementers, we did also add greater transparency and clarity into the processes and best practices we have in place for collaboration and upstream contribution to the platform.
- **Payment Hub Documentation**: As the use cases for which Payment Hub continue to evolve, documentation efforts will continue to grow; through this project we have refreshed documentation to try to make payment hub more accessible for end users.
- **Functional & Deployment Documentation:**Another major milestone that we accomplished was updating our user manual with brand new screenshots to reflect the new user interface in the web-app that has replaced our deprecated Community App.
- **Standardize Documentation**: We have underestimated the remaining effort that is still outstanding to consolidate and streamline our user and developer documentation so this is still an going effort to ensure that developers, users, and implementers can all efficiently access the documentation required to complete their desired task.

Project Results

Pull Requests for documentation submitted to Fineract ACSII doc can be found here: https://github.com/apache/fineract/pulls/tapaswenipathak

Phase	Description	Deadline	Result
Standardize Documentation			
● Define standard for existing documentation and ensure each project's user and technical documentation adheres to that standard format/flow:	Design Outlines for ● docs.mifos.org ● GitHub/OpenMF	6 June, 2022	● https://github.com/openMF/mifos-documentation/issues/15 ● https://github.com/openMF/android-client/issues/1877 ● https://github.com/openMF/mifos-documentation/issues/17
● For end-users - installation, troubleshooting common issues, system specs, tutorials, product overview, quick start guide, domain/topic explainers. ● For developers - contributing guidelines, project setup guide, troubleshooting tips, architectural description/overview, release notes/history.	Provide recommendations on how documentation can seamlessly exist w/o duplications across more than 7 documentation platforms and reduced entry points: 1. Landing Page: https://mifos.org/resources/docs 2. User Documentation Portal: https://docs.mifos.org 3. Technical Documentation Portal [Developer/Contributor Docs]: https://mifos.readme.io 4. Apache Wiki 5. Fineract Documentation [ASCII doc]: https://fineract.apache.org/docs/current/ 6. User/Implementer Docs: https://mifos.gitbook.io/docs/ 7. Mifos Wiki 8. OpenMF GitHub Wiki	6 June, 2022	● https://github.com/openMF/android-client/issues/1877 ● https://github.com/openMF/android-client/issues/1876 ● https://github.com/openMF/android-client/issues/1875 ● https://github.com/openMF/mobile-wallet/issues/1332 ● https://github.com/openMF/mobile-wallet/issues/1331 ● https://github.com/openMF/mifos-mobile-cn/issues/229 ● https://github.com/openMF/mifos-mobile/issues/1863 ● https://github.com/openMF/mifos-documentation/issues/18 ● https://github.com/openMF/mobileapps.github.io/issues/1
Organize existing documentation according to proposed information	● Efforts are ongoing now to consolidate the new documentation written in our ASCII doc and wikis under the user and developer documentation portals.		
Architectural Documentation			

Refine documentation on the enhanced ability to separate read/write API operations for greater performance	● Refinement to existing documentation on the enhanced ability to separate read/write/batch API operations for greater performance ● Add customer documentation for read/write/batch API operation performance enhancing separation	10 October 2022	● Customer Documentation ● Enhanced examples
Refine documentation for migration of database from MySQL to PostgreSQL including the migration from Flyway to Liquibase for database migration scripts	● Refine documentation for migration of database from MySQL to PostgreSQL including the migration from Flyway to Liquibase for database migration scripts	10 October 2022	● FAQs ● Migration to 1.8 ● Enhanced examples
Refine documentation for tuning and indexing to optimize performance based on PostgreSQL and Eclipselink enhancements.	● Refine documentation for tuning and indexing to optimize performance based on PostgreSQL and Eclipselink enhancements.	10 October 2022	● FAQs ● Migration to 1.8 ● Enhanced examples ● Performance improvements ● Extensibility documentation
Write new documentation on transaction idempotency and business and effective date configuration	● Add business date customer document ● Add customer documentation for effective date	10 October 2022	● Customer Documentation ● Release documentation
Refine documentation on new reliable event handling framework built on Kafka	● Refine documentation on new reliable event handling framework built on Kafka	10 October 2022	@Ed C
Refine documentation on integration of Spring Batch for parallelization of batch jobs, running smaller chunks of batch jobs, ensuring no transaction conflict between real-time and batch processes including architecture of new framework, how to configure and schedule jobs, etc.	● Refine documentation on integration of Spring Batch for parallelization of batch jobs	10 October 2022	● Documentation of feature ● Customer Documentation ● Release Documentation
Functional & Deployment Documentation			
Draft new documentation outlining functionality of payment hub, each of its new modules, and use cases it supports	https://github.com/openMF/mifos-documentation/issues/14#issuecomment-1305471505	7 Nov, 2022	● 2022 Payment Hub

Draft new documentation on how to install, deploy and configure Payment Hub EE	● https://mifos.gitbook.io/docs/payment-hub-ee/overview/installation-instructions ● https://mifos.gitbook.io/docs/payment-hub-ee/overview/deployment-models ● https://github.com/openMF/mifos-documentation/issues/14#issuecomment-1305469459	7 Nov, 2022	● Enhancements to installation ● FAQs ● Specifics
Update documentation and screenshots in user manual to reflect design of new web app	● https://github.com/openMF/mifos-documentation/issues/14#issuecomment-1309069264	28 Nov, 2022	● Updated https://sandbox.mifos.community/ for each screen present in https://docs.mifos.org/mifosx/user-manual/for-all-users ● Updated every textual description on all pages listed https://github.com/openMF/mifos-documentation/issues/14#issuecomment-1309069264 ● Created tickets/issues/bugs for missing functionality and others https://github.com/openMF/mifos-documentation/issues/14
Update and refine documentation to reflect new DevOps tooling for deployment of the platform and Payment Hub EE including Terraform scripts, Configurable Helm Charts, Kubernetes and Containerization Optimations	● https://github.com/openMF/mifos-documentation/issues/14#issuecomment-1305466568	7 Nov, 2022	● Added documentation
Update critical documentation that is outdated or not clear including installation guide, reporting, SMS gateway configuration, and self-service user creation.	● Still ongoing as new features are in development and not yet ready to be documented from user perspective.		
Collaboration Documentation			
Draft documentation related to automated code checks, static analysis, and other build-level automation that has been introduced to improve code quality and reduce friction of upstream contribution	Add Collaboration Documentation	31 October, 2022	● Added documentation
Update documentation on upstream contribution process and ensuring that contributors know how to properly configure Github to sync their downstream codebase with upstream project and do	Add Collaboration Documentation	31 October, 2022	● Added documentation

development in an upstream-first manner that is aligned with Apache Way			
Draft documentation for QA/testers on how to write new tests and leverage the Cucumber Test framework	Add documentation for Cucumber Test Framework ● Additional documentation still forthcoming to list out the Cucumber tests that have been written and shared with community.	2 Nov, 2022	● Further enhanced documentation ● Fixed todos in documentation
Draft documentation for developers on how to write code leveraging new Cucumber test framework for unit testing	Add documentation for Cucumber Test Framework	2 Nov, 2022	● Further enhanced documentation ● Fixed todos in documentation
Update FAQs explaining difference between Mifos and Fineract.			● Enhanced FAQs ● Per mailing list ● GitHub issues ● GitHub discussions ● Slack ● Other channels
Review and Incorporate Feedback			
Review all of the deliverables			
Case Study & Publication			
Case Study		28 November 2022	
Publications	Pull Requests will soon be merged and part of the release documentation of each new Fineract release. Changes to user manual on Gitbooks are already reflected and live. Changes to other documentation like within the wiki are being updated in coordination with the ASCII doc release.	@Ed C	

Results

Updated and added documentation on all the available source of information absorption by users of Mifos, which are:

1. mifos.org
2. Developer/Contributor Docs - https://mifos.readme.io/docs/getting-started (readme.io) (Technical, how to build, contribute, extend)
3. Apache wiki
4. ASCII doc - https://fineract.apache.org/docs/current/
5. User/Implementer Docs - docs.mifos.org (gitbook) (More around use case, business need, functionality), https://mifos.gitbook.io/docs/
6. Mifos Wiki
7. OpenMF GitHub wiki

Measuring Project Success

Through workshops and surveys we continually hear from the community how we need to reduce the friction in the developer experience. We also have a dearth in qualified senior developers that we hope to scale up through better documentation.

We would consider the project successful if, after updating documentation to support the launch of Mifos X 3.0:

- Number of pull requests and upstream contributions increase by 20%
- Number of onboarded new active contributors increase by 10 over a 3 month period.
- Number of partners making contributions upstream increases by at least 5 additional partners.
- Number of questions on mailing list regarding to how deploy or configure the platform goes down by 20%

Baseline metrics are starting to be gathered now and performance against these metrics will be tracked on a quarterly basis.

Analysis & Learnings

- **Strong Project and Product Management Skills:** Having a technical writer with a rich product management background like that of Tapasweni was valuable in adding structure and discipline to the documentation process and having a high attention to detail around the documentation, its purpose and what value it will provide. Understanding of the product lifecycle and the proactive
- **Burden on Subject Matter Experts -** during the previous Season of Docs, getting access to subject matter experts was a challenge that we did overcome by ensuring early and proactive access. Nonetheless, an ongoing challenge that we are still trying to overcome is reducing the burden upon the subject matter expert when they have to participate in multiple calls to effectively provide the requisite background information or source materials. It becomes a delicate balance in determining if it would be more time-efficient to

have the developer or subject matter expert write the documentation directly rather do all the knowledge transfer.

- **Domain Knowledge is Precious:** Despite the strong product development mindset, it still proved somewhat challenging at times to have our technical writer be as effective as possible without as much functional domain knowledge regarding financial services and financial inclusion and/or technical domain knowledge around enterprise core banking architecture as that of an active and current community member. Having a deeper baseline of domain knowledge could probably help alleviate some of the burden on subject matter experts.
- **Outlining and Defining Goals/Purpose of Documentation is Key**. Early on we learned the importance of for each piec
- **Working in Parallel to Active Development:** While working on documentation for something that's actively being developed has its benefits in terms of fresh source materials and access to subject matter experts, it can pose a challenge if some of the deliverables aren't yet complete and ready for public consumption. If the development of features are also customer-facing, it can also be challenging to ensure technical writers is aware of customer-senstive information that can't be included in public documentation.
- **Documentation as Code:** Having everything written in markdown format and shared as either pull requests to our Gitbook or pull requests to our ASCII doc as part of the upstream codebase made for easier review and collaboration and ongoing updates. However having documentation pull requests as part of the general flow of upstream pull requests sometimes became distracting to our developers and could lead to stagnant pull requests that don't get merged in a timely fashion
- **Added Friction on top of Developer Documentation:** This point goes back to that of subject matter experts but when we have existing documentation that a developer has written and are having our technical writer polish, refine and add more substance to it, it becomes another careful and delicate balance with the additional value this adds versus simply having the developer themself refine and finalize the documentation. Sometimes the purpose or intent or specific details can get lost in the refinement process.

Appendix

Appendix A: Folder Sources

1. Google Drive GSoD'22 References:
    a. https://docs.google.com/document/d/1x_ucMp8O3fWwwtBCdLm_IwsO31LLg5C96LJugR-Ryec/edit#heading=h.845tlywdunr2

2. [Tracker] Season of Docs - 2022: Links/urls: docs, planners, tickets, PRs: https://github.com/openMF/mifos-documentation/issues/14

Appendix B: Docs Tools

1. **Google Docs** for Documenting the docs.
2. **ClickUp** for Project Management Tool.
3. **Readme.io**
4. **gitbook.io**
5. **Apache Wiki**
6. **GitHub**
7. **Draw.io**

Appendix C: Project Resources

1. Landing Page: https://mifos.org/resources/docs
2. User Documentation Portal: https://docs.mifos.org
3. Technical Documentation Portal [Developer/Contributor Docs]: https://mifos.readme.io
4. Apache Wiki
5. Fineract Documentation [ASCII doc]: https://fineract.apache.org/docs/current/
6. User/Implementer Docs: https://mifos.gitbook.io/docs/
7. Mifos Wiki
8. OpenMF GitHub Wiki