

HEPCon – A Cross-Platform Mobile Application for HEP Events

Martin Vassilev^{1,*}, *Vassil Vassilev*^{2,**}, *Alexander Penev*^{1,***}, and *Petya Vassileva*^{3,****}

¹Plovdiv University Paisii Hilendarski, 236 Bulgaria blvd, Plovdiv, Bulgaria

²Princeton University, Princeton, New Jersey 08544, United States

³CERN, Meyrin 1211, Geneve, Switzerland

Abstract. Collaboration in research is essential for saving time and money. The field of high-energy physics (HEP) is no different. The higher level of collaboration the stronger the community. The HEP field encourages organizing various events in format and size such as meetings, workshops and conferences. Making attending a HEP event easier leverages cooperation and dialogue and this is what makes Indico service defacto a community standard.

The paper describes HEPCon, a cross-platform mobile application which collects all information available on Indico and makes it available on a portable device. It keeps most of the data locally which speeds up the interaction. HEPCon uses a shared code base which allows easy multiplatform development and support. There are iOS and Android implementations available for free download. The project is based on C# and we use the Xamarin mobile app technology for building native iOS and Android apps. SQLite database is responsible for retrieving and storing conference data. The app can be used to preview data from past CHEP conferences but the tool is implemented generic enough to support other Indico events.

1 Introduction

Many conferences in the High-Energy Physics (HEP) community are centralized and organized through the Indico platform. Indico is an open source project designed for handling the management of HEP events varying in sizes – from small meetings and workshops towards complex scientific conferences with hundreds of participants. Indico has proven itself with more than 210000 events organized [1].

The motivation for creating HEPCon is to simplify event organization and participation and to overcome the problems with the persistent internet connection and the mobile device compliant presentation of data. With HEPCon various information from Indico with respect to the organization and attendance of a conference is collected and put on a mobile device for offline use.

*e-mail: mvasilev@uni-plovdiv.bg

**e-mail: vvasilev@cern.ch

***e-mail: apenev@uni-plovdiv.bg

****e-mail: petya.vasileva@cern.ch

The key feature of HEPCon is the instant synchronization with the Indico service. When a change in Indico is made all the agenda data could be automatically downloaded by HEPCon and stored in its memory.

This Indico synchronization helps organizers, because other conference apps have their own data format and all data should be migrated every time. This is especially tedious when it comes to last minute changes. When HEPCon downloads the information from both data sources it caches it. This allows the user to have all necessary information available without internet connection. This is handy because the user could download the data when there is a free wifi connection and use the app while offline. Another advantage of HEPCon is that it supports both major platforms - Android and iOS.

2 Background

The flexibility of Indico in terms of providing solid backend for conference organization is further supported by being adopted by various international organizations. Key capabilities which makes Indico the standard system for providing support for scientific events are:

- Full life cycle coverage - providing support from attendee registration and abstract submission to modules for paper reviewing;
- Integration with external tools;
- Rest API.

Indico is a web based system and as such it inherits the problems of every web system – the requirement for an active internet connection which would allow its users to interact with it and the recently introduced requirement for a dedicated data presentation when viewed on mobile devices.

The process of entering conference related data in a central platform like Indico is time consuming. For example, for CHEP2018 there are nearly 76 sessions and more than 570 contributions. If the conference organizers decide to overcome the above mentioned problems using a standard mobile application they would have to double their efforts when maintaining conference data since a change in Indico would cause the administrators to make a change in the mobile application backend.

Pathable, Guidebook, Attendify, Whova and Eventbase are well known products [2]. Unfortunately, they do not have Indico integration and force organizers to double their efforts when making changes related to the conference data. While duplicating information may not be a concern for small events, it can be devastating for larger events.

As a result of the research for the availability of similar solutions a conclusion there are numerous event related mobile applications on the market could be made. However, while they provide feature rich solutions for various types of events they lack the support of Indico. In addition, often the solutions on the market are expensive.

3 Architecture

The architecture of HEPCon has strict separation between the user interface and the business and presentation layers.

When apps are being developed and new features are added complex maintenance problems could arise. These problems are more likely to appear if a tight coupling between the user interface elements and the business logic is present. When an architecture with a tight coupling between the UI and login layers is present the cost of making modifications is increased.

The Model-View-ViewModel (MVVM) design pattern enables developers to decouple the user interface from the business and presentation layers. Maintaining a clean separation between application logic and the UI helps to address numerous development issues and can make an application easier to test, maintain, and evolve [3].

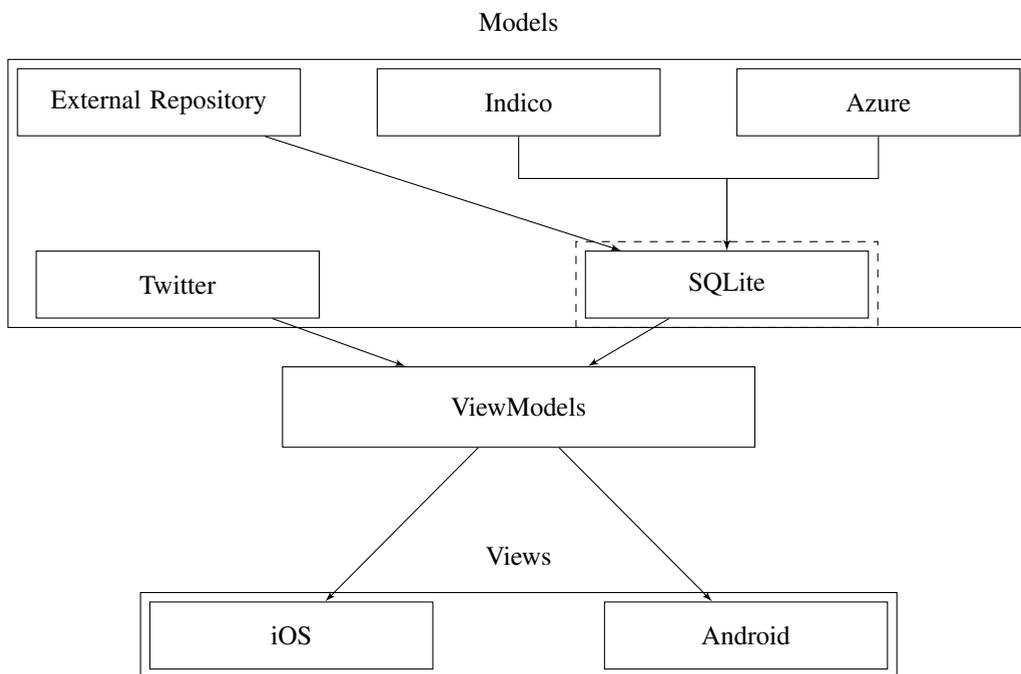


Figure 1. HEPCon architecture schema.

Figure 1 represents the architecture of HEPCon. There are four models available each responsible for encapsulating data from different endpoints. The data coming from Indico, Azure and the external repository is cached via SQLite database which increases data access speed and offline use.

The views from Figure 1 are responsible for specifying how the user interface should look like. Using XAML the developer have only to describe the visual elements and their properties once in shared files while Xamarin handles the translation to native user interface controls.

The viewmodels implement the middle layer whose main responsibility is the decoupling between data and views. This is achieved using properties and commands to which the view can bind to. Event notification mechanisms are used for communication between the layers.

4 Implementation

Indico itself is not designed to provide all the information necessary for a conference attendee. This is the reason HEPCon has an integration with an external repository which holds the rest of the standard conference information such as organization committees; important venues; conference venues; hotels; official dinner; sponsors; code of conduct and venue floor maps. The external repository could be used to decouple the otherwise strict coupling in terms of

exchanging data between Indico and HEPCon. One of the advantages of having another kind of data repository, besides Indico, is that HEPCon could display custom data that is not strictly related with the Indico format. The support for creating announcements available almost immediately in HEPCon is implemented by using this additional data infrastructure.

The success of every software project depends on several factors. The main factors are the correct choices of programming language and technologies to use when designing the application. In addition, HEPCon had to conform to additional, often mutually exclusive, constraints such as limited time for implementation and limited funding. Finding an open source project to use as foundation was crucial for the successful delivery of the application.

The research for an open sourced project with good architecture was successful – the Xamarin Evolve 2016 app [4] was found which is open sourced and provided very good architectural design and very good looking interface. The project is based on the Xamarin technology which allows to have a shared code base for multiple platforms and only override the platform-specific features. Using this method there is roughly more than 90% shared code base and less than 10% platform specific code.

For example, the data manipulation code is shared while only the main pages of the app are platform specific since iOS supports the concept of the MultiPage while Android – the so called Master/Details concept. The rest of the UI pages are also shared.

4.1 Technology Stack For Cross-Platform Development

HEPCon is based on well-known technologies designed for multiplatform applications development.

4.1.1 C#

C# is a relatively new programming language when compared with Objective-C and Java [5]. At first, C# seemed to be a rather straightforward, strongly typed, imperative object-oriented language, certainly influenced by C++ (and Java as well), but with a much cleaner syntax than C++ and none of the historical baggage [6]. C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions [7].

4.1.2 Xamarin

With a C#-shared codebase, developers can use Xamarin tools to write native Android, iOS, and Windows apps with native user interfaces and share code across multiple platforms, including Windows and macOS [8].

Xamarin focused mainly on compiler technologies and basic sets of .NET libraries:

- Xamarin.Mac, which has evolved from the MonoMac project;
- Xamarin.iOS, which evolved from MonoTouch;
- Xamarin.Android, which evolved from Mono for Android or (more informally) MonoDroid.

Collectively, these libraries are known as the Xamarin platform. The libraries consist of .NET versions of the native Mac, iOS, and Android APIs. Programmers using these libraries can write applications in C# to target the native APIs of these three platforms, but also (as a bonus) with access to the .NET Framework class library [5].

4.1.3 SQLite

SQLite is an embedded database. Rather than running independently as a stand-alone process, it symbiotically coexists inside the application it serves—within its process space. Its code is intertwined, or embedded, as part of the program that hosts it. To an outside observer, it would never be apparent that such a program had a relational database management system (RDBMS) on board. The program would just do its job and manage its data somehow, making no fanfare about how it went about doing so. But inside, there is a complete, self-contained database engine at work [9].

4.2 Data Model

In general the HEP events are composed from series of contributions or sessions running in sequential order or in parallel. The data coming from the Indico model from Figure 1 is responsible for handling the Indico agenda. This agenda have a strong relation with time and as such the model responsible for it could be considered as a time based representation.

This time based representation could be seen as a timeline [10, 11] using the model's contributions or sessions starting date times as a base and the current moment as a reference point for the exact moment the timeline state is analyzed.

Having a timeline based model influences the format of the visualization of the agenda data. The view have to handle the timeline no matter whether there are sequential or parallel sessions or contributions. HEPCon provides functionality for filtering sessions by date time.

4.3 Native Look And Feel

HEPCon data is separated in multiple pages. Figure 2 and Figure 3 represents the iOS and Android versions.

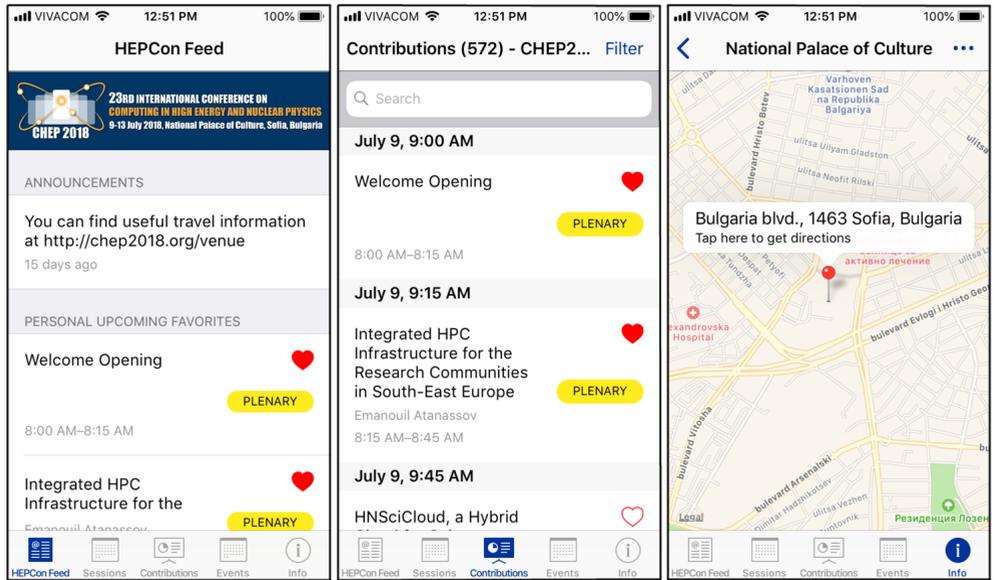


Figure 2. HEPCon iOS screenshots. The Feed, Contribution, Points of Interest and Venue pages are shown.

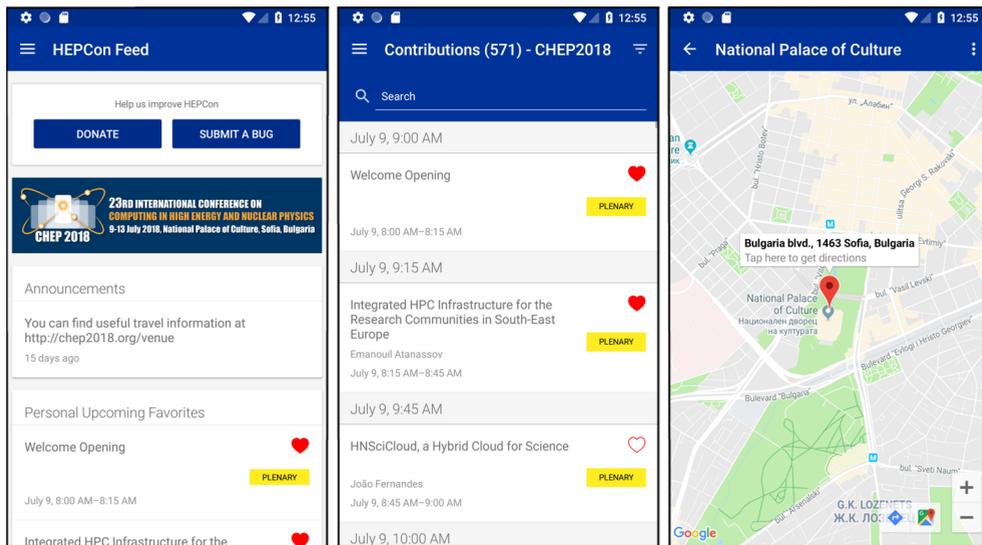


Figure 3. HEPCon Android screenshots. The Feed, Contribution, Points of Interest and Venue pages are shown.

4.3.1 Feed

The first page the user would see when HEPCon is installed is the Feed page. It is the place where the user could obtain information related to the general aspects of a conference like:

- Announcements - important messages issued by the conference organizers;
- Favored contributions - a list of user selected contributions available to remind the user of their upcoming presentation;
- Ongoing contributions - a list of contributions or events happening at the current moment;
- Twitter feed.

4.3.2 Sessions

The Sessions page contains all the Indico tracks and other important entities like registration and breaks. The tracks could be described as aggregated contributions from the timetable based on their corresponding venue and time slots. Every session contains a list of contributions related to the corresponding track. A menu is created for filtering sessions based on the day of the conference.

4.3.3 Contributions

The third page is dedicated to the Contributions list. On this page the information like contributions title, when and where it will be held could be found. The user would be able to see who is the speaker and if she or he has more contributions, they will be listed when the speakers name is tapped.

If the user marks the contribution as a favorite, then it will be listed on the Feed page. This way, the user could keep track of the talks he or she is interested in.

The user could add the contribution to its calendar which could turn a reminder on. The user could do filtering, based on date time, whether to show only favorites or to list only contributions from specific tracks.

There is a rating system implemented and every user could give a rating to a contribution. A rating button will appear when the contribution is finished. Currently, the ratings are only for personal use and kept locally on the phone.

Various filters are available on the page. The user is allowed to filter by title or whether to show past contributions as of the current moment. The user could choose to show only contributions from the a specified track or to show only the favorites.

4.3.4 Events

The Events page is the place where the user could load a different conference. This page provides a list with the conferences available to be loaded in HEPCon. To avoid hardcoding values the list of the conferences is pulled from the external repository. This list could be edited without having to upload to the App Store and Google Play new versions of the app. The list shows essential information for every conference available like the title, location and date time.

Every conference have its own details page. On this page the user could find more details about the conference and the button to load it in HEPCon. When new conference is loaded the local cache of HEPCon is cleared and the data of the new conference is downloaded. The rest of the pages like the Feed, Sessions, Contribution, Info, are automatically updated with the new information.

4.3.5 Info

The Info Page is the place that holds all the other relevant information. The first part of the page is related with providing information when was the last sync and a syncing button to force refresh HEPCon with the data from Indico and the external repository.

The rest of the page displays information for:

- Conference committees - lists of members of IAC and LOC;
- Sponsors - a branded list of sponsors with tags to show the level of sponsorship;
- Evaluations - a list of contributions with user ratings;
- Points of interest - a list of important venue locations. Each venue is related to a Google or Apple map where the Get directions functionality could be activated;
- Conference floor maps - a list of images indicating the halls of the conference center;
- Code of conduct;
- Wi-Fi information;
- Submit a bug button.

5 Future Work

One of the most important things planned for implementation is to provide the users with the ability to load arbitrary conference without having them to open a pull request to the external repository. The user would enter the Indico address of the conference and HEPCon could fetch the data.

The tedious process of submitting conference related documents like travel requests, receipts and others could be handled by taking photos through the app. This would greatly improve the HEP conferences experience. OCR functionality could be integrated as well so that the ease of manipulation of the transmitted data could be enhanced.

Push notifications could be integrated since it is a better way to provide announcements. Currently, the announcements are handled through the external repository which is not very convenient.

Currently, the personalization of the app is not possible. A login functionality would allow to create a unified rating system for the contributions. The login functionality would allow HEPCon to provide the ability for participants to create chat rooms. Those chat rooms could be used by the attendees only interested in certain topic which would increase the ease of collaboration between them.

Other plans include automatic info merge, auto select conference to load based on the venue and date time.

Conveners view is another feature which was requested by HEPCon users. It would allow while a speaker is in the middle of a talk to monitor the time remaining so she or he do not run out of time. This will also give a real time information of the progress of the sessions.

HEPCon is also open for consideration to include other public data sources different from Indico.

References

- [1] J.B.G. Lopez, *Indico 1.0*, Journal of Physics: Conference Series **513** (2014)
- [2] *The best event apps for every type of event*, <https://www.eventbrite.com/blog/best-event-apps-ds00/>
- [3] D. Britch, *Enterprise Application Patterns using Xamarin.Forms* (Microsoft Press, A Division of Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399, 2017)
- [4] *Xamarin evolve 2016 app development retrospective*, <https://montemagno.com/xamarin-evolve-2016-app-development-retrospective/>
- [5] C. Petzold, *Cross-platform C# programming for iOS, Android, and Windows* (Microsoft Press, A Division of Microsoft Corporation, One Microsoft Way, Redmond, Washington 98052-6399, 2016)
- [6] R. Miles, *C# Programming Yellow Book* (University of Hull, Hull, 2016)
- [7] *C sharp (programming language)*, [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [8] *Xamarin*, <https://en.wikipedia.org/wiki/Xamarin>
- [9] M.O. Grant Allen, *The Definitive Guide to SQLite* (Apress, 2010)
- [10] S.F. Silva, T. Catarci, *Visualization of linear time-oriented data: a survey*, in *Web Information Systems Engineering, 2000. Proceedings of the First International Conference on* (IEEE, 2000), Vol. 1, pp. 310–319
- [11] D. Rosenberg, A. Grafton, *Cartographies of time: A history of the timeline* (Princeton Architectural Press, 2013)